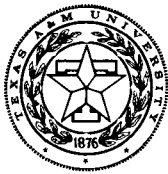


JOHNSON
GRANT
11-34-012
200056
652P.

COMPUTATIONAL FLUID DYNAMICS
AND AEROTHERMODYNAMICS



aerospace engineering department

Final Report
January 1987 -- December 1988

TEXAS A&M UNIVERSITY

TAMRF Report No. 5671-89-01
March 1989

NASA Grant No. NAG 9-192
TAMRF Project 5671

(NASA-CR-180079) COMPUTATIONAL FLUID
DYNAMICS AND AEROTHERMODYNAMICS Final
Report, Jan. 1987 - Dec. 1988 (Texas A&M
Univ.) 652 F CSCI 20D

N89-20415

Unclas
0200056
G3/34

Leland A. Carlson
Professor of Aerospace Engineering
Texas A&M University
College Station, TX. 77843-3141

TEXAS AEROSPACE ENGINEERING EXPERIMENT STATION

COMPUTATIONAL FLUID DYNAMICS
AND AEROTHERMODYNAMICS

Final Report

January 1987 -- December 1988

TAMRF Report No. 5671 - 89 - 01

March 1989

NASA Grant No. NAG 9-192
TAMRF Project RF 5671

Leland A. Carlson
Professor of Aerospace Engineering
Texas A&M University
College Station, TX 77843-3141

TABLE OF CONTENTS

I.	Introduction.....	Page 3
II.	Personnel.....	3
III.	Accomplishments.....	3
IV.	Conclusion.....	10
V.	Appendix 1 -- AIAA Papers.....	14
	AIAA Paper No. 88-2672, Carlson, L. A., "Approximations for Hypervelocity Nonequilibrium Radiating, Reacting, and Conducting Stagnation Regions"	
	AIAA Paper No. 88-2673, Carlson, L. A., Bobskill, G. J., and Greendyke, R. B., "Comparison of Vibration Dissociation Coupling and Radiative Heat Transfer Models for AOTV/AFE Flowfields"	
VI.	Appendix 2 -- Master's Thesis of Glenn J. Bobskill.....	40
	"A Comparative Study of Reaction Rate, Species, and Vibration-Dissociation Coupling Models for an AOTV Flowfield"	
VII.	Appendix 3 -- Compendium of Reaction Rate, Species, and Vibration-Dissociation Coupling Results.....	163
VIII.	Appendix 4 --Master's Thesis of Robert B. Greendyke.....	307
	"A Parametric Study of Shock Jump Chemistry, Electron Temperature, and Radiative Heat Transfer Models in Hypersonic Flows"	
IX.	Appendix 5 -- Program RAPROX.....	404
X.	Appendix 6 -- The AFE Programs.....	429

COMPUTATIONAL FLUID DYNAMICS AND AEROTHERMODYNAMICS

(Development of Nonequilibrium Models Applicable to AOTV/AFE and
Superorbital Flight Regimes)

I. Introduction

This report summarizes the activities and accomplishments associated with Texas A&M Research Foundation Project 5671 which was funded as a subportion of NASA Grant No. NAG 9-192 from the NASA Johnson Space Center. This project was awarded January 6, 1987 and actively continued until December 6, 1988. The primary objective of this effort was the development of nonequilibrium radiation and chemistry models suitable for engineering applications associated with the flowfields about aero-assisted orbital transfer vehicles (AOTVs), the aero-assisted flight experiment vehicle (AFE), and other vehicles operating at superorbital velocities and very high altitudes.

II. Personnel

While the project was officially awarded in January 1987, the fiscal paperwork was not completed for several months and actual work was not permitted to start until mid-spring 1987. However, since that date did not correspond to the beginning of an academic semester, the graduate students planned for the project were not immediately available. In late spring, Mr. Glenn J. Bobskill was assigned to the project as a research assistant; and in June 1987 Mr. Robert B. Greendyke joined the team, also as a research assistant. Both students officially remained with the project thru May 1988. Both Mr. Greendyke and Mr. Bobskill used their research work on the project as the basis for their master's theses; and both received their M.Sc. degrees in August 1988. In addition, Mr. Thomas Gally, who is a doctoral student, worked on the project at various times. At the present time, Mr. Gally is continuing the work initiated on this project on a NASA Graduate Student Researcher Fellowship from the NASA Johnson Space Center.

Originally, the work proposed for this effort was only supposed to last for one year. However, since paperwork delayed the effective start of the project until June 1987, it was immediately extended until June 1988. Subsequently, in order to allow the presentation of results at the AIAA Thermophysics, Plasmadynamics, and Lasers Conference in June 1988 and to permit the development of final versions of various computer codes, reports, etc., the project was extended until December 1988. However, from a funding and accomplishments standpoint, almost all work on the project was completed before the end of August 1988.

The principal investigator for this project has been Dr. Leland A. Carlson, Professor of Aerospace Engineering.

III. Accomplishments

In this section the primary accomplishments and achievements of this project will be briefly discussed and summarized. In most cases, complete details are available in the various appendices of this report.

The primary results of this project are contained in the following publications:

AIAA Paper No. 88-2672: Carlson, L. A., "Approximations for Hypervelocity Nonequilibrium Radiating, Reacting, and Conducting Stagnation Regions," Presented at the AIAA Thermophysics, Plasmadynamics, and Lasers Conference, June 27-29, 1988, San Antonio, Texas. Accepted for publication in the Journal of Thermophysics and Heat Transfer.

AIAA Paper No. 88-2673: Carlson, L. A., Bobskill, G. J., and Greendyke, R. B., "Comparison of Vibration Dissociation Coupling and Radiative Heat Transfer Models for AOTV/AFE Flowfields", Presented at the AIAA Thermophysics, Plasmadynamics, and Lasers Conference, June 27-29, 1988, San Antonio, Texas. Scheduled for publication in the Journal of Thermophysics and Heat Transfer in October 1989.

Master's Thesis: Bobskill, Glenn James, "A Comparative Study of Reaction Rate, Species, and Vibration-Dissociation Coupling Models for an AOTV Flowfield," Master of Science Thesis, Aerospace Engineering, Texas A&M University, College Station, Texas, August 1988.

Master's Thesis: Greendyke, Robert Brian, "A Parametric Study of Shock Jump Chemistry, Electron Temperature, and Radiative Heat Transfer Models in Hypersonic Flows," Master of Science Thesis, Aerospace Engineering, Texas A&M University, College Station, Texas, August 1988.

These AIAA papers and theses are contained in Appendices 1, 2, and 4 of this report.

In addition the following paper, which is based upon work initiated under the present project, will be presented next summer:

Carlson, L. A. and Gally, T. A., "The Effect of Electron Temperature and Impact Ionization on Martian Return AOTV Flowfields," To be presented at the AIAA 24th Thermophysics Conference, June 12-15, 1989, Buffalo, New York.

Besides these publications, several computer programs have been developed as part of this project. In general, these codes can be classified as either being approximate or detailed. The approximate code, called RAPROX, is designed to provide rapid flowfield and heat transfer estimates for the stagnation region of vehicles in the 11-18 km/sec range, and thru a series of approximations is designed to include the effects of thermal conduction, chemical nonequilibrium, and nonequilibrium nongray radiative transfer. Predictions obtained with the method show reasonable agreement with the Fire 2 flight experiment and with the results of other investigators (Ref. 1). However, parametric studies conducted with the method indicate that further studies of vacuum ultra-violet (VUV) processes and radiation models is needed. This method is discussed in detail in AIAA Paper 88-2672 (Ref. 2) contained in Appendix 1. Further, user instructions, a listing, and sample results, are presented in Appendix 5.

The detailed code is an inviscid nonequilibrium chemistry axisymmetric inverse code based upon the work of Grosse (Ref. 3). However, it has been extensively

modified to include a variety of vibration-dissociation chemistry options; and models for radiative nonequilibrium effects, electron temperature, and radiative heat transfer computations have been developed and included. Both microcomputer and mainframe versions of the code have been created. Fortunately, or unfortunately, depending upon viewpoint, this code has experienced continual change as new phenomena were discovered and models developed. Consequently, several different versions of the code currently exist. These various versions, which differ primarily in reaction chemistry details and electron temperature models, are presented and discussed in Appendix 6.

The primary results obtained with this code were presented in Ref. (4), which is reprinted in Appendix 1, and in the results presented in Appendices 2 thru 4. The primary conclusions were:

- (1) At the exit trajectory point for the AFE, simplified reaction schemes can yield useful results at significant computational savings, providing that ionized species are not of interest.
- (2) A modified coupled vibration dissociation vibration coupling scheme (MCVDV) was shown to yield reasonable vibrational and electron temperature profiles as well as species concentration variations.
- (3) At entry and Max Q trajectory points for the AFE, detailed reaction systems are required in order to properly predict the charged species affecting the radiative heat transfer. Further, the results are sensitive to the electron impact ionization rate; and further study to properly model ionization chemistry is needed.
- (4) Nonequilibrium significantly affects the radiative heat transfer, and both molecular and atomic nonequilibrium radiation correction factors are essential for proper prediction. However, radiative cross-sections and absorption coefficient models based on equilibrium results can be used provided they are corrected for nonequilibrium radiation phenomena.
- (5) The treatment of line radiation at the low density AFE conditions needs further study and refined radiation models suitable for engineering calculations need to be developed.

As indicated above, the primary efforts on this project were completed during the summer of 1988. However, a low level of effort has been continued by the principal investigator and several interesting developments have occurred since the last progress report.

First, results obtained with both the approximate model and the detailed code indicated that of the radiation cross-section models studied, the eight step model due to Olstad (Ref. 5) was the best. In addition, during the summer of 1988 a copy of the detailed radiation transport model/code RADICAL, which was developed by NASA Langley and based upon Ref. (6), was obtained. Consequently, comparison studies between the eight-step model and RADICAL were conducted. These studies revealed that the cross-sections predicted by the eight-step model for the wavelength region 1130-1801 angstroms, which predominantly contains atomic

lines, were high. Subsequent investigation indicated that Olstad had multiplied his original values for this region by a factor of 20 to account for revised f numbers. However, examination of the various f numbers revealed that perhaps an average increase of eight would have been more appropriate. Thus, the Olstad values for the region in question were divided by 2.5 and new comparisons with RADICAL computed. These new results agreed very well with RADICAL, and thus the radiation models in RAPROX and the AFE codes were appropriately modified.

Second, during the fall of 1988 an electron temperature model based upon quasi-equilibrium locally was developed and a preliminary version incorporated into several of the AFE codes as an option. This new model determines the free electron temperature and takes into account both elastic and inelastic collision effects, including electron impact ionization. Thus, the new codes actually are "three-temperature" in that they predict heavy particle or translational temperature, free electron temperature, and vibrational temperatures for each diatomic species. The versions discussed in Ref. (4) assumed that the electron temperature was equal to the nitrogen vibrational temperature. It is believed that this new model will be more valid than the previous version at velocities above 11.5 km/sec, where free electrons dominate the ionization chemistry and the radiation.

Third, the validity of various electron impact ionization rates and models is in the process of being studied. In particular, solutions have been obtained using various proposed rates along with various electron temperature models. In addition, the ionization model in the programs is in the process of being modified to account for a two step process. In other words, it is assumed that ionization proceeds via collisional excitation to an excited state followed by subsequent rapid ionization. The preliminary results indicate significantly different flowfield predictions can result based upon which model is used. It is anticipated that these studies will be presented at the next AIAA Thermophysics Meeting along with comparisons of existing experimental data.

Fourth, in the MCVDV vibration dissociation coupling model used in the AFE codes the vibrational relaxation time predicted by Millikan and White type of correlations is corrected to prevent unrealistically short relaxation times, i.e.

$$\tau = \tau_{MW} + \frac{1}{N C \sigma_v}$$

where $C = \text{SQRT}(8KT/\pi m)$ and according to Park (Ref. 7)

$$\sigma_v = (1E-17) \left[\frac{50000}{T} \right]^2$$

is a good correlation for the vibrational excitation cross section. For the nonequilibrium region immediately near the shock front, this expression will yield values around $5E-17$. However, Park (Ref. 8) and others (Gnoffo, private

communication) have previously used values as high as $1\text{E}-16$ for this cross section. For the AFE flight regime, these differences can change the vibrational relaxation time by an order of magnitude and significantly change the vibrational temperature and subsequently the electron temperature and chemistry profiles. In turn, the radiative heat transfer predictions can also be varied significantly. Consequently, a series of preliminary studies have been conducted to investigate this effect. Specifically, results have been obtained for several trajectory points in which the leading coefficient in the G_v equation has been either $1\text{E}-17$ or $1\text{E}-16$.

Fifth, in spite of the effort in the present project, there is still considerable uncertainty concerning the application and use of the present nonequilibrium radiation model, or for that matter any nonequilibrium radiation model, in the AFE flight regime. At higher velocities, say 12 km/sec , where the degree of ionization is large (i.e. electron mole fractions around 0.20), there are sufficient electrons so that electron impact ionization dominates the chemistry and that a computed free electron temperature has meaning. Under such conditions, the presence of thermal and chemical nonequilibrium should significantly affect populations in excited electronic states and the nonequilibrium radiation correction factors developed in the present effort should be appropriate and adequate.

However, at the AFE max Q trajectory point (8.915 km/sec , $15.715\text{ dynes/sq cm}$, 197.101 K), the mole fraction of electrons is less than one percent. Under these conditions there might be insufficient electrons for the excited states of atoms to be in equilibrium with the ions and free electrons at the local free electron temperature, and it might be more likely that the temperature characterizing the electronic excited states of both atoms and molecules would be closer to the nitrogen vibrational temperature due to strong vibrational-electronic coupling. This type of coupling is essentially what was assumed in Ref. (4) when T_e was assumed equal to T_{vN_2} . Further, it might be reasonably assumed that the atomic electronic states would be populated according to a Boltzmann equilibrium distribution at the vibrational temperature. It should be noted that this is not what was assumed in Ref. (4), where atomic and molecular nonequilibrium correction factors were both used.

This latter possibility is borne out by some detailed nonequilibrium radiation results presented by Park (Ref. 9) for a condition similar to the max Q point. His results indicate that the excited electronic states of nitrogen, particularly those near the $3s4P$ level, essentially follow a Boltzmann distribution at the assumed electron temperature. On the other hand, Park's corresponding predictions for the nonequilibrium excitation of N_2 molecules shows that the molecular electronic states are not populated according to a Boltzmann distribution and that typically the numbers in each excited state are one or more orders of magnitude lower than that predicted by a Boltzmann distribution at T_e . Thus, for engineering calculations with methods like the AFE codes, it might be reasonable to assume at the max- Q point that $T_e = T_{vN_2}$, that the electronic levels of the atomic species are in Boltzmannian equilibrium (no atomic correction factors), and that molecular species are in radiative nonequilibrium (include molecular correction factors).

The consequences of these uncertainties in reaction chemistry, vibration-dissociation cut-off values, electron temperature models, and radiative nonequilibrium models, significantly affects the predictions of radiative heat transfer at trajectory points such as the max-Q points. Predicted values, all obtained using the eight step radiative cross section model, are shown on Table I for various assumptions.

TABLE I -- Results Obtained for Various Models
U = 8.915 km/sec, p infinity = 15.715 dynes/sq cm, 9 cm above stag. pt.

Case	Te Model	v	Rates	Aton	Rad	Mol. Rad	VUV	UV+Lines	Vis+IR	Total
1	TWN2	1E-16	RR3	Noneq	Noneq		.33	.27	.52	1.12
2	"	"	"	Equil	Noneq		3.43	2.32	2.80	8.55
3	"	"	"	Noneq	Equil		2.74	4.17	53.27	60.18
4	"	"	"	Equil	"		5.15	6.21	55.54	66.90
5	QE	"	"	Noneq	Noneq		.32	.21	.31	.85
6	"	"	"	Equil	Noneq		79.61	41.53	40.16	161.30
7	"	"	"	Noneq	Equil		28.90	55.94	450.70	535.60
8	"	"	"	Equil	Equil		91.27	96.56	490.00	677.90
9	TWN2	1E-17	Wilson	Noneq	Noneq		.03	.03	.05	.10
10	"	"	"	Equil	Noneq		.36	.28	.52	1.16
11	"	"	"	Noneq	Equil		.21	.13	5.45	5.79
12	"	"	"	Equil	Equil		.48	.38	5.92	6.77
13	"	1E-16	"	Equil	Noneq		4.17	2.84	3.40	10.40
14	QE	1E-17	"	Noneq	Noneq		.03	.02	.03	.08
15	"	"	"	Equil	Noneq		39.06	22.19	23.83	85.08
16	"	"	"	Noneq	Equil		14.80	23.99	263.20	302.00
17	"	"	"	Equil	Equil		45.98	45.87	286.70	378.60
18	"	1E-16	"	Noneq	Noneq		.02	.02	.03	.07

NOTES

All radiative heating rates in watts/sq cm.

Number in 3rd column is leading constant in correlation for v.

QE -- T_e determined by quasi-equilibrium free electron temperature.

TWN2 -- T_e assumed equal to T_{vN2} .

RR3 -- Reactions and rates from Ref. (4).

Wilson -- Oxygen ionization reaction in RR3 replaced by $O+e=O^++2e$

Rates for both electron impact nitrogen and oxygen ionization are based upon experiments of Wilson (Ref. 10).

The species concentrations and vibrational temperatures used for the cases marked "RR3" are shown on Fig. 17 of Ref. (4), while those for cases denoted "Wilson" are shown on Figs. (1) and (2) in this report. In particular, notice on Figs. (1) and (2) the significant differences in the vibrational temperature profiles due to the differences in vibrational excitation cross section (1E-16 vs. 1E-17 leading coefficient). While not as obvious, the 1E-16 case also exhibits faster relaxation and lower translational temperatures at corresponding locations due to the shorter vibrational relaxation time and corresponding higher vibrational temperatures, which enhance dissociation. In addition, these results were all obtained on a microcomputer using only 20 streamlines. While in many

cases that number is adequate, to be conservative these values should not be considered "definitive" but should be used for comparison purposes. Further, since the AFE codes are inviscid, the radiative heating values should be viewed as the flux to the edge of the wall viscous layer. Since a considerable portion of the VUV and UV radiation should be absorbed in the "cool" viscous layer, the above values should be conservative. However, very little of the visible plus IR will be absorbed in the viscous layer.

At first glance the results presented in Table I appear to be confusing, particularly considering the wide range of values presented. However, by comparing various cases the importance of individual phenomena can be determined and insight into the uncertainty associated with various facets of the problem can be ascertained. For example, the sensitivity to reaction systems and rates can be determined for this trajectory point by comparing case (2) with (13) and case (1) with (18). These indicate that if atomic electronic states are in Boltzmanian equilibrium then the radiation is not particularly sensitive to atomic ionization chemistry. On the other hand, if radiative nonequilibrium exists for both atomic and molecular species [1 vs 18], significant sensitivity exists, although both predictions from these latter two cases are fairly small.

Likewise, the sensitivity to the method of computing the electron temperature used in the radiation calculations can be observed by comparing cases (1-4) with cases (5-8) respectively and (9-12) with (14-17). These results show that if any of the electronic states, either molecular or atomic, are in Boltzman equilibrium at T_e that the radiative heating predictions are very sensitive to the T_e model; although, if radiative nonequilibrium is assumed for atoms and molecules, the sensitivity to electron temperature modeling is considerably less.

Similarly, the dependence of the results on the leading constant in the vibrational excitation correlation can be estimated by examining cases (10) and (13) for the situation where atomic levels are assumed to be in radiative equilibrium. These two results show about an order of magnitude difference, indicating strong dependence upon vibrational and vibration-dissociation coupling phenomena. On the other hand, cases (9) and (18) state that when radiative nonequilibrium is considered for all species, that the vibrational excitation cross section effect is less and in the opposite direction.

By comparing case (1) with (2) and (9) with (10), the effect of assuming radiative equilibrium or nonequilibrium in atomic species can be observed for the situation where $T_e = T_{VN2}$; and the results indicate that atomic radiative equilibrium leads to approximately a 1-7 watts/sq cm increase in the radiative heating predictions, depending upon the vibrational excitation cross section. However, when T_e is determined by a quasi-equilibrium electron temperature model, cases (5) and (6) and (14) and (15) show that the existence of radiative equilibrium among atomic states has the potential of increasing heating predictions by 80 to 160 watts/sq cm, indicating extreme sensitivity to electron temperature modeling.

Likewise, the sensitivity to molecular radiative nonequilibrium is indicated by cases (1) vs (3) and (9) vs (11) for $T_e = T_{VN2}$ and by (5) vs (7) and (14) vs (16) for T_e from quasi-equilibrium. In the latter case the existence of radiative

equilibrium for molecular species leads to extremely large, and hopefully "incorrect", radiative heating increases of 300 to 500 watts/sq cm; while in the former case assuming molecular radiative equilibrium leads to smaller increases ranging from 5 to 60 watts/sq cm.

Considering the wide range of predictions depending upon models and assumptions selected the question arises Which prediction is the "best guess"? As previously indicated, at the max-Q trajectory point the amount of ionization is very low and there is evidence that atomic electronic states may be near Boltzman equilibrium while molecular electronic states could have significant radiative nonequilibrium. In addition, for this case temperatures over much of the shock layer are fairly low (8000 - 10,000 K) and electronic vibrational coupling should be significant. In fact, Ref. (11) states that the vibration-electronic relaxation time for N₂ is smallest (and hence coupling the greatest) at 7000 K. Further, this author intuitively favors the vibrational temperature profiles for cases where the vibrational excitation cross section leading coefficient is 1E-16. Thus, the "best guess" cases for the max-Q trajectory point might be cases (2) and (13). Interestingly, the radiative heating predictions in the visible and infrared from these two cases is in reasonable agreement with the results of Ref. (11) when it is realized that the present predictions are for a location nine cm above the axis. Of course, this "agreement" may be serendipity, especially when it is recognized that cases (11) and (12) predict about the same values.

At higher velocities, the above assumptions would not be as correct and in many cases would definitely be incorrect. By 12 km/sec, the amount of ionization would be significant, free electrons would dominate the ionization chemistry and electronic level excitation, and dissociation would be very rapid. In those cases, it probably would be better to assume radiative nonequilibrium for both atomic and molecular species and to use as the electron temperature the value predicted by an appropriate free electron equation model.

IV. Conclusion

As demonstrated above and in the appendices, considerable progress has been made under the present grant in the development of nonequilibrium chemical and radiative engineering models for AOTV/AFE high altitude flight regimes. However, the present effort has also demonstrated that radiative heating predictions for a typical AFE vehicle are very sensitive to the existence or nonexistence of radiative nonequilibrium among the atomic electronic states and to the values predicted for electron temperature. Thus, additional research work is needed in the area of electron electronic energy equation modeling; and current engineering approaches to modeling atomic radiative nonequilibrium phenomena need to be refined. In addition, it would be desirable to establish both experimentally and theoretically the existence and degree of molecular radiative nonequilibrium at AFE conditions since molecular radiative equilibrium leads to extremely high heating rate predictions.

References

1. Park, C. and Davies, C. B., "Aerothermodynamics of Manned Mars Mission," AIAA Paper 89-0313, January 1989
2. Carlson, L. A., "Approximations for Hypervelocity Nonequilibrium Radiating, Reacting, and Conducting Stagnation Regions," AIAA Paper No. 88-2672, June 1988.
3. Grosse, W. L. "A Thin Shock Layer Solution for Nonequilibrium Inviscid Flows in Earth, Martian, and Venusian Atmospheres," NASA TN D-6529, December 1971.
4. Carlson, L. A., Bobskill, G. J., and Greendyke, R. B., "Comparison of Vibration Dissociation Coupling and Radiative Heat Transfer Models for AOTV/AFE Flowfields", AIAA Paper No. 88-2673, June 1988.
5. Olstad, W. B., "Nongray Radiating Flow about Smooth Symmetric Bodies," AIAA Journal, Vol. 9, No. 1, Jan. 1971, pp. 122-130.
6. Nicolet, W. E., "Advanced Methods for Calculating Radiation Transport in Ablation Product Contaminated Boundary Layers," NASA CR-1656, Sept. 1970.
7. Park, C. , "Assessment of Two-Temperature Kinetic Model for Ionizing Air," AIAA Paper No. 87-1574, June 1987.
8. Park, C., "Problems of Rate Chemistry in the Flight Regimes of Aeroassisted Orbital Transfer Vehicles," in Vol. 96 of Progress in Astronautics and Aeronautics, H. F. Nelson, ed., AIAA, New York, N.Y., 1985, pp. 511-537.
9. Park, C., "Calculation of Nonequilibrium Radiation in the Flight Regimes of Aeroassisted Orbital Transfer Vehicles," in Vol. 96 of Progress in Astronautics and Aeronautics, H. F. Nelson, ed., AIAA, New York, N.Y., 1985, pp. 395-418.
10. Wilson, J., "Ionization Rate of Air Behind High Speed Shock Waves," Physics of Fluids, Vol. 9, No. 10, October 1966, pp. 1913-1921.
11. Candler, G. and Park, C., "The Computation of Radiation from Nonequilibrium Hypersonic Flows," AIAA Paper No. 88-2678, June 1988.

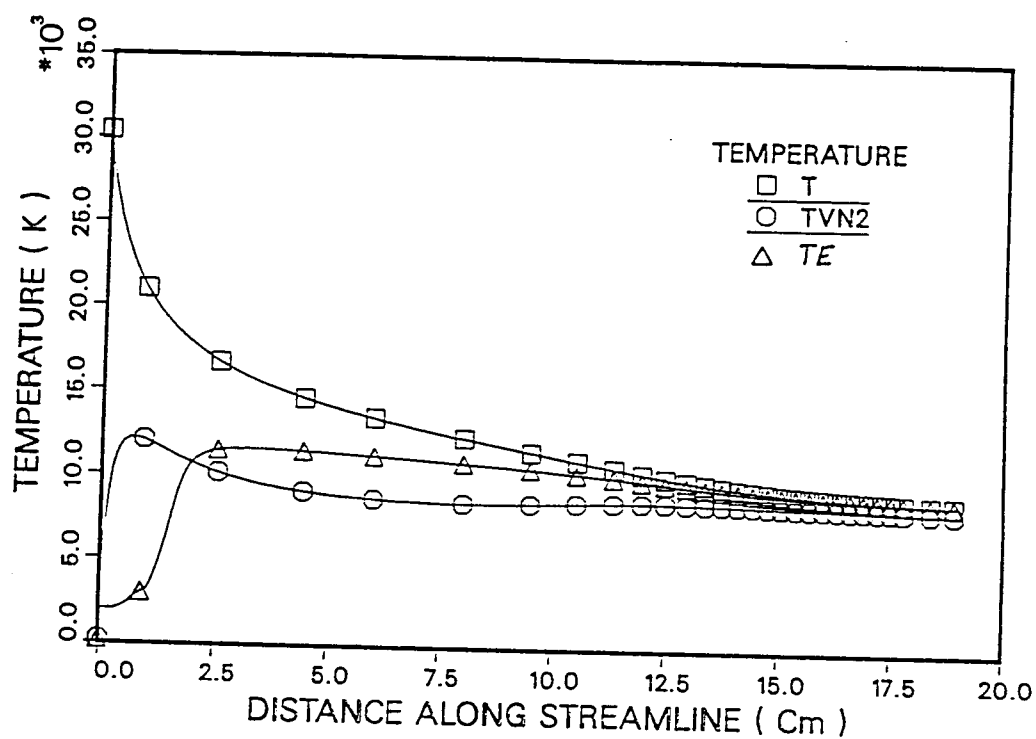
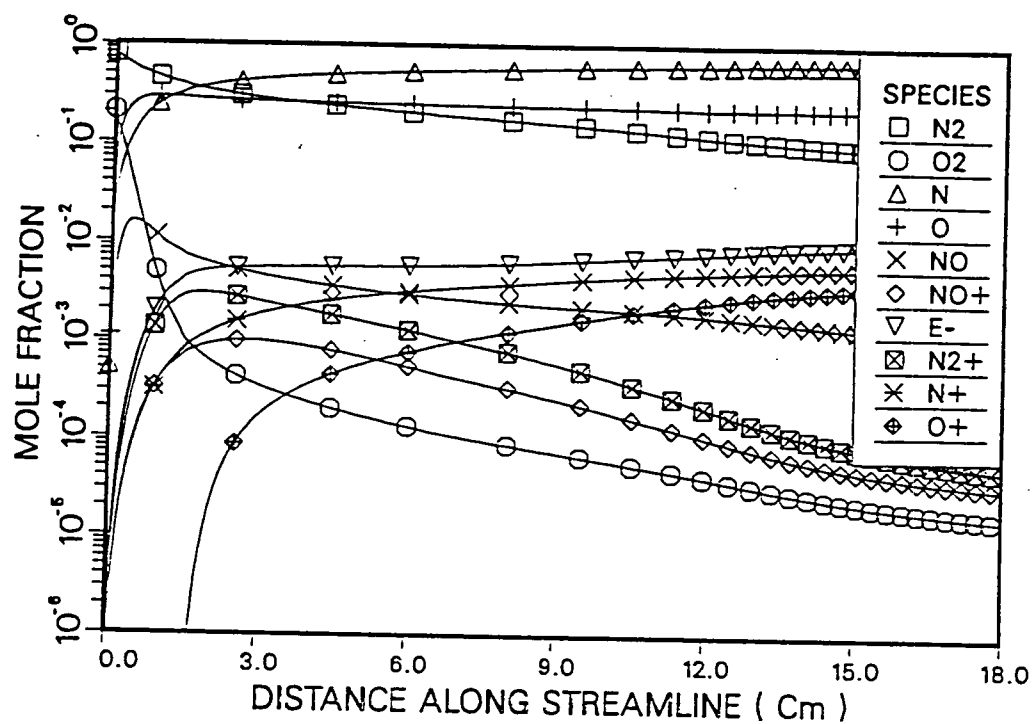


Fig. 1 Species and Temperature Profiles for MCV DV
Coupling, $G_v \sim 10^{-16}$ U = 8.9 Km/sec

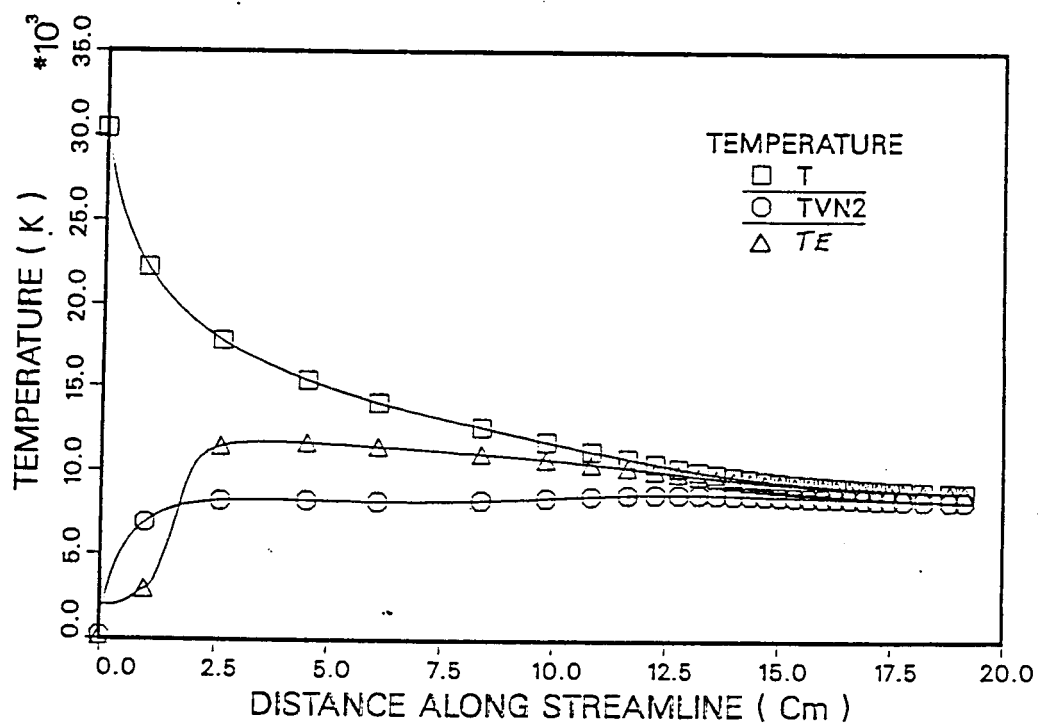
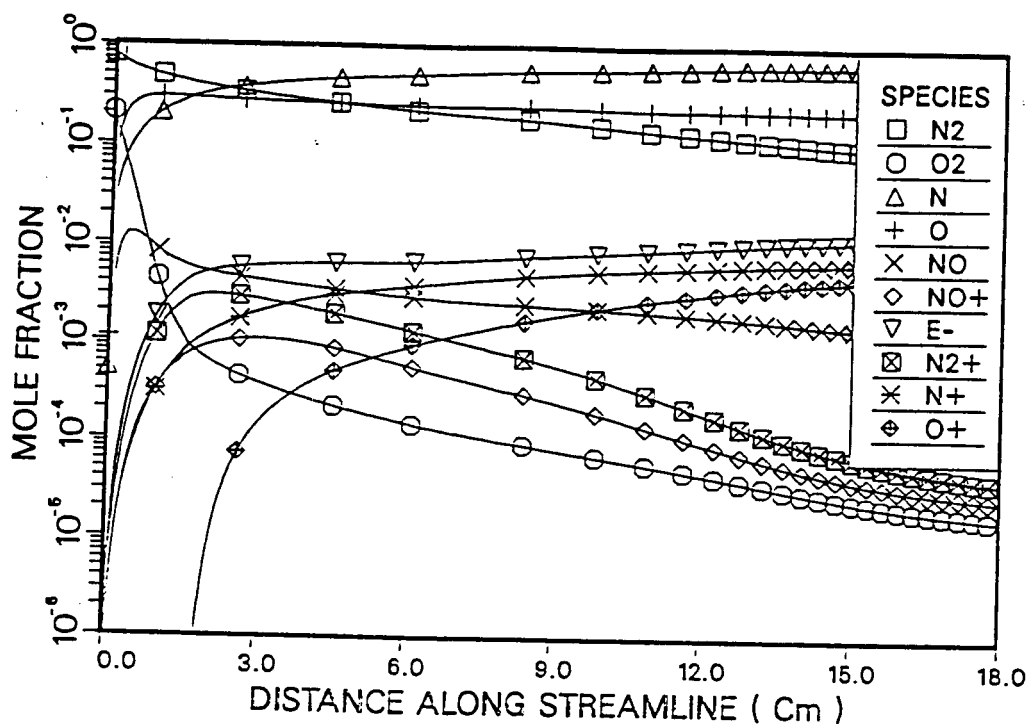


Fig. 2 Species and Temperature Profiles for MCVDV
Coupling, $G_V \sim 10^{-17}$, $U = 8.9$ Km/sec

APPENDIX 1

AIAA Papers

88A43740 AIAA Paper No. 88-2672, Carlson, L. A., "Approximations for Hypervelocity Nonequilibrium Radiating, Reacting, and Conducting Stagnation Regions"

88A43741 AIAA Paper No. 88-2673, Carlson, L. A., Bobskill, G. J., and Greendyke, R. B., "Comparison of Vibration Dissociation Coupling and Radiative Heat Transfer Models for AOTV/AFE Flowfields"

A COMPARATIVE STUDY OF REACTION RATE, SPECIES, AND VIBRATION
-DISSOCIATION COUPLING MODELS FOR AN AOTV FLOWFIELD

A Thesis

by

GLENN JAMES BOBSKILL

Submitted to the Graduate College of
Texas A&M University
in partial fulfillment of the requirements for
the degree of

MASTER OF SCIENCE

August 1988

Major Subject: Aerospace Engineering

A COMPARATIVE STUDY OF REACTION RATE, SPECIES, AND VIBRATION
-DISSOCIATION COUPLING MODELS FOR AN AOTV FLOWFIELD

A Thesis

by

GLENN JAMES BOBSKILL

Approved as to style and content by:

Dr. Leland A. Carlson
(Chairman of Committee)

Dr. Cyrus Ostowari
(Member)

Dr. David L. Rhode
(Member)

Dr. Walter E. Haisler
(Head of Department)

August 1988

ABSTRACT

A Comparative Study of Reaction Rate, Species, and Vibration-
Dissociation Coupling Models for an AOTV Flowfield

(August 1988)

Glenn James Bobskill, B.S., Texas A&M University

Chairman of Advisory Committee: Dr. Leland A. Carlson

An analysis was conducted using various combinations of reaction rates, species, and vibrational coupling models at three different AOTV trajectory points. A comparative study of the CVD, CVDV, CVDV-Preferential, and Park-Like vibrational coupling models combined with three reaction rate sets were used to determine the best modeling combinations. The conclusions obtained were that the results from the CVD model were considered to be incorrect due to the model's inability to compensate for the effects that dissociation and recombination have on vibration, that additional reaction rate studies were needed in order to ascertain the validity of the electron impact rate coefficients, that the Park-Like model produced a better representation of electron temperature over a wider velocity range than the other models, and the use of a smaller reaction rate set will produce reasonable results with a substantial reduction in computational time.

DEDICATION

To the two women in my life who mean more to me than my
future

To my mom,

whose love, guidance, and gentle thoughts
helped me when I needed it mosteveryday.....

To my wife to be,

although we've had our problems, I feel warm
and secure in the knowledge that we'll face life
.....together.....

ACKNOWLEDGEMENTS

I wish to first thank the members of my advisory committee for their participation and patience. Special thanks go to Dr. Leland Carlson for his guidance, knowledge and patience in dealing with our performances on a daily basis. I considered it a privilege to have had the opportunity to work with him. I also wish to thank Robert Ratcliff for his expertise in IBM computer modifications and DISSPLA tutorials. In addition, I wish to thank Derek Schrock for his valuable assistance in thesis editing and organization.

This work was primarily supported under a subportion of NASA Grant No. NAG 9-192 from NASA Johnson Space Center, with Dr. Carl Scott, Aerosciences Branch as technical monitor. Computational support was also provided by the College of Engineering and the Office of the Associate Provost for Computing and Information Systems.

TABLE OF CONTENTS

	Page
ABSTRACT	iii
DEDICATION	iv
ACKNOWLEDGEMENTS	v
TABLE OF CONTENTS	vi
LIST OF TABLES	vii
LIST OF FIGURES	viii
NOMENCLATURE	xi
INTRODUCTION	1
BACKGROUND	5
CHEMICAL KINETICS	16
SOLUTION METHOD AND ANALYSIS	21
SOLUTION PROCEDURE	27
INTERNAL ENERGY MODEL	36
RESULTS AND DISCUSSION	39
Reaction Rate Set 1	42
Reaction Rate Set 2	57
Reaction Rate Set 3	74
Reaction Rate Studies	91
CONCLUSIONS	105
REFERENCES	106
APPENDIX	108
VITA	110

LIST OF TABLES

Table	Page
1. AOTV Trajectory Points	40
2. Reaction Rate Set 1	40
3. Reaction Rate Set 2	40
4. Reaction Rate Set 3	41

LIST OF FIGURES

Figure		Page
1.	Generic AOTV Configurations	2
2.	Vibrational Relaxation Data for Nitrogen . . .	15
3.	Shock-Oriented Coordinate System	22
4.	Flow Chart of Solution Algorithm	35
5.	Veq model at V=10 Km/s, RR1	44
6.	CVD model at V=10 Km/s, RR1	45
7.	CVDV model at V=10 Km/s, RR1	46
8.	CVDV-P model at V=10 Km/s, RR1	47
9.	Park-L model at V=10 Km/s, RR1	48
10.	T_{vn2} and N_2 Profiles at V=10 Km/s, RR1	49
11.	O_2 and N Profiles at V=10 Km/s, RR1	50
12.	O and NO Profiles at V=10 Km/s, RR1	51
13.	NO^+ and e^- Profiles at V=10 Km/s, RR1	52
14.	Coordinates, V=10 Km/s, RR1	56
15.	Veq model at V=10 Km/s, RR2	59
16.	CVD model at V=10 Km/s, RR2	60
17.	CVDV model at V=10 Km/s, RR2	61
18.	CVDV-P model at V=10 Km/s, RR2	62
19.	Park-L model at V=10 Km/s, RR2	63
20.	T_{vn2} and N_2 Profiles at V=10 Km/s, RR2	64
21.	O_2 and N Profiles at V=10 Km/s, RR2	65
22.	O and NO Profiles at V=10 Km/s, RR2	66

LIST OF FIGURES (CONTINUED)

Figure		Page
23.	NO^+ and e^- Profiles at $V=10$ Km/s, RR2	67
24.	N_2^+ and N^+ Profiles at $V=10$ Km/s, RR2	68
25.	O^+ Profile at $V=10$ Km/s, RR2	69
26.	Coordinates, $V=10$ Km/s, RR2	73
27.	Veq model at $V=10$ Km/s, RR3	75
28.	CVD model at $V=10$ Km/s, RR3	76
29.	CVDV model at $V=10$ Km/s, RR3	77
30.	CVDV-P model at $V=10$ Km/s, RR3	78
31.	Park-L model at $V=10$ Km/s, RR3	79
32.	$T_{\text{vn}2}$ and N_2 Profiles at $V=10$ Km/s, RR3	80
33.	O_2 and N Profiles at $V=10$ Km/s, RR3	81
34.	O and NO Profiles at $V=10$ Km/s, RR3	82
35.	NO^+ and e^- Profiles at $V=10$ Km/s, RR3	83
36.	N_2^+ and N^+ Profiles at $V=10$ Km/s, RR3	84
37.	O^+ Profile at $V=10$ Km/s, RR3	85
38.	Coordinates, $V=10$ Km/s, RR3	90
39.	$T_{\text{vn}2}$ and N_2 Profiles at $V=10$ Km/s, CVDV	92
40.	O_2 and N Profiles at $V=10$ Km/s, CVDV	93
41.	O and NO Profiles at $V=10$ Km/s, CVDV	94
42.	NO^+ and e^- Profiles at $V=10$ Km/s, CVDV	95
43.	N_2^+ and N^+ Profiles at $V=10$ Km/s, CVDV	96
44.	O^+ Profile at $V=10$ Km/s, CVDV	97

LIST OF FIGURES (CONTINUED)

Figure	Page
45. T_{vn2} and N_2 Profiles at $V=7.7$ Km/s, CVDV	99
46. O_2 and N Profiles at $V=7.7$ Km/s, CVDV100
47. O and NO Profiles at $V=7.7$ Km/s, CVDV101
48. NO^+ and e^- Profiles at $V=7.7$ Km/s, CVDV102
49. N_2^+ and N^+ Profiles at $V=7.7$ Km/s, CVDV103
50. O^+ Profile at $V=7.7$ Km/s, CVDV104

NOMENCLATURE

C	- Molecular velocity
c_i	- Mass fraction of species i
\bar{E}	- Vibrational energy lost during single dissociation
F°	- Free energy at standard pressure
\bar{G}	- Vibrational energy gained during single recombination
h°	- Heat of formation at standard pressure
h	- Enthalpy
k_b	- Backward reaction rate coefficient
k_f	- Forward reaction rate coefficient
M	- Molecular specie
N	- Number of vibrational energy levels to dissociation
N_O	- Avagadro's number
n	- Number density
p	- Pressure
Q	- Partition function
R	- Universal gas constant
R_c	- Radius of curvature
T	- Translational temperature
T_a	- Geometric average temperature
T_m	- $TT_v/(T + T_v)$
T_v	- Vibrational temperature
U	- Characteristic probability temperature
u, v	- Component of velocity parallel and normal to shock

NOMENCLATURE (CONTINUED)

α	- Coefficient used in vibrational relaxation time
β	- Temperature exponent, vibrational relaxation time
ϵ_v	- Vibrational energy
θ_v	- Characteristic vibrational temperature
λ	- Scale factor $(1 - \gamma)/R_c$
μ	- Chemical potential
μ_{m_i}	- Molecular weight of species i
ν	- Stoichiometric coefficient
ρ	- Mass density
σ	- Coefficient used in vibrational relaxation time
σ_v	- Collisional cross section
τ	- Vibrational relaxation time
τ_c	- Corrected vibrational relaxation time
ϕ	- Coupling factor
ψ	- Stream function
ω_n	- Rate of production of species n
$\omega_e \begin{Bmatrix} x_e \\ y_e \\ z_e \end{Bmatrix}$	- Anharmonic oscillator coefficients

Subscripts:

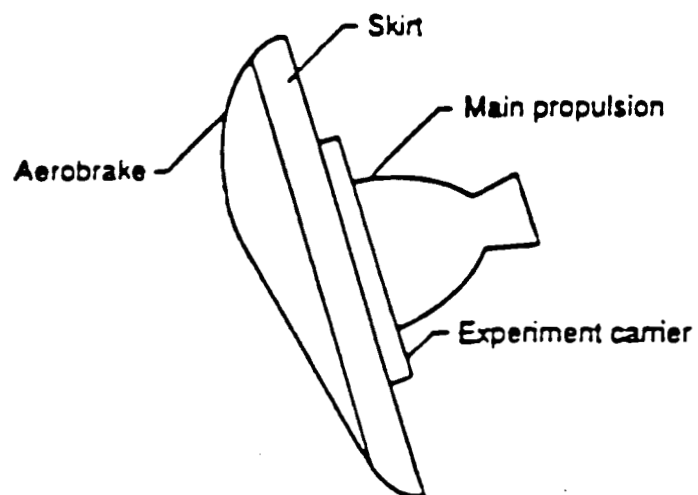
b	- Body
s	- Shock
∞	- Refers to local vibrational equilibrium or freestream

INTRODUCTION

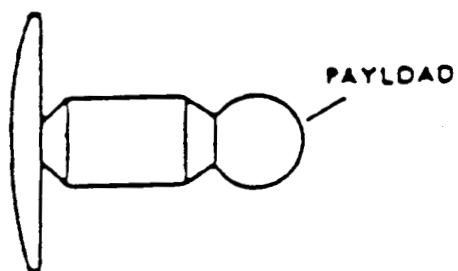
In recent years, there has been a renewed interest within the aerospace industry to predict the chemical and thermodynamic quantities in a hypersonic flowfield. At present, the Aeroassisted Orbital Transfer Vehicle (AOTV) represents one of the current and future applications of hypersonic flowfield research within the aerospace community. The AOTV performs an aerobraking maneuver, using atmospheric drag to reduce kinetic energy, to enable the vehicle to transition from a higher geosynchronous orbit to a lower earth orbit without expending available energy resources. From a practical standpoint, the use of this device would decrease the cost and increase the payload size of future missions in space. Some examples of various AOTV configurations are displayed in Figure 1. The velocity and altitude range normally encountered during a geosynchronous to a lower earth orbit maneuver is about 7 to 11 kilometers per second at 70 to 100 kilometers. Furthermore, the maximum dynamic pressure point (Q_{\max} or $\max-Q$) encountered during this maneuver occurs at an altitude of approximately 75 kilometers.

These mission requirements, at their respective altitudes and velocity ranges, have made apparent the need to

Format in accordance with the AIAA Journal



(a)



(b)

FIGURE 1. GENERIC AOTV CONFIGURATIONS
(Adapted from References 1 and 2)

develop methods which would yield good engineering results and at the same time be computationally efficient. One technique which fulfills these requirements is an inviscid method developed by Maslen³. Maslen's approach involves the solution of the momentum and energy equations, within the transformed von Mises plane, as functions of a given the x shock coordinate and a stream function. The Maslen method is an inverse formulation in which the shock shape is mathematically specified in a two dimensional coordinate system and the subsequent flow parameters and body shape are then determined as part of the final solution. Later, Grose⁴ extended the Maslen type approach by including chemical and thermal nonequilibrium effects and applying them to various planetary atmospheres.

The rate at which chemical relaxation occurs behind a strong shock wave strongly affects on all chemical and thermodynamic quantities in a high speed flowfield. These chemical relaxation processes are governed primarily by chemical reaction rates and the change in vibrational energy for each diatomic species component during collisional exchanges. This process of chemical-kinetic interaction displays a continuous energy exchange between the vibrational modes of motion, the dissociation, and recombination processes that take place within the region of relaxation. Vibration-dissociation coupling models are used to describe these energy exchanges. Thus, they significantly affect the determination of the correct species concentration,

vibrational temperature, and other thermodynamic quantities.

The objective of the research, as outlined within this thesis, was to perform a comparative study of various species, reaction rate models, and vibration-dissociation coupling models within the proposed AOTV flight trajectories. In addition, an analysis was performed to determine which combinations of species, chemical reaction rates, and vibrational coupling models were necessary to properly describe the chemical and vibrational relaxation processes for the AOTV. The program listed in Reference 4 was modified to perform this study.

BACKGROUND

It has been revealed through previous experimental studies that the processes of vibrational relaxation, dissociation, and recombination do not occur independently; but rather, each process influences the other through continuous energy exchanges. Vibration-dissociation coupling models which are used in an attempt to describe this kinetic process are fundamentally based on the dissociation of a pure diatomic gas as derived from thermochemical concepts. Several past and present investigations have produced viable models which are included within the present comparative study.

The first model used in this analysis is the Coupled-Vibration-Dissociation⁵ model (CVD). The CVD model is comprised of essentially two mathematical expressions which are

$$\frac{\partial \epsilon_v}{\partial t} = \frac{\epsilon_{v_\infty} - \epsilon_v}{\tau} \quad (1)$$

$$\phi = \frac{1}{N} \left\{ \frac{1 - e^{\left[-N \left(\frac{\theta_v}{T_v} - \frac{\theta_v}{T} \right) \right]}}{e^{\left(\frac{\theta_v}{T_v} - \frac{\theta_v}{T} \right)} - 1} \right\} \left\{ \frac{e^{\left(\frac{\theta_v}{T_v} \right)} - 1}{e^{\left(\frac{\theta_v}{T} \right)} - 1} \right\} \quad (2)$$

Equation (1) describes the rate at which the vibrational energy changes due to collisions and also assumes that the molecules can be represented as a system of simple harmonic oscillators. This equation describes the rate of change of vibrational energy at any instant as being linearly proportional to the amount that the vibrational energy differs from equilibrium at that instant. It should be noted that no assumption was made concerning how the oscillators were distributed over the available energy levels in the original derivation of equation (1). In particular, the standard assumption of a Boltzmann distribution was neglected. The Boltzmann distribution assumes that at sufficiently high temperatures, the particles are distributed over a wide range of permissible energy levels and that the distributions themselves can be described in the form of partition functions.

Equation (2) is referred to as the coupling coefficient or coupling factor which couples the effects of vibrational nonequilibrium to the dissociation process. The coupling factor is defined as the ratio of the actual forward rate to the forward rate that would exist for local vibrational equilibrium. The final form of equation (2) comes about from the redefinition of the forward rate ratio in the form of partition functions. The substitution of the vibrational partition function into this manipulation redefines the Morse energy curve into a system of harmonic oscillators cutoff just prior to dissociation. The Morse curve is a

representation of changes in vibrational energy as a function of intermolecular distance. As the distance changes, due to vibration, work must be done against the interatomic forces. These forces are attractive at large distances and repulsive at short distances. The CVD model also assumes that dissociation can occur with equal probability from any vibrational level. Details of this derivation are outlined in the Appendix. It should also be noted that the program given in Reference 4 already incorporates the CVD model and therefore required no additional modification.

The second model that has been used in this study is the Coupled-Vibration-Dissociation-Vibration^{6,7} model (CVDV). The vibrational rate equation for the CVDV model is comprised of the following terms

$$\frac{\partial \epsilon_v}{\partial t} = \frac{\epsilon_{v_\infty} - \epsilon_v}{\tau} - \left\{ \left[\frac{\theta_v}{e^{\left(\frac{\theta_v}{T_m}\right)} - 1} - \frac{N\theta_v}{e^{\left(\frac{N\theta_v}{T_m}\right)} - 1} \right] - \epsilon_v \right\}$$

$$\frac{1}{n} \left[\frac{\partial n}{\partial t} \right]_f + \left[\frac{1}{2} (N - 1) \theta_v - \epsilon_v \right] \frac{1}{n} \left[\frac{\partial n}{\partial t} \right]_r \quad (3)$$

The CVDV model is similar to the CVD model to the extent that they share the same basic assumptions and formulation, except that equation (3) combines the vibrational rate equation of the CVD model plus two additional terms. These additional terms take into account the effect that dissociation has on

the average vibrational energy of molecules as previously represented by a system of simple harmonic oscillators which are cutoff just prior to dissociation. The second term, the forward process, in equation (3) represents the average energy lost from vibration due to a single dissociation, while the third term, the reverse process, represents the amount of average energy gained due to a single recombination.

These additional terms were derived based on the assumption that all vibrational levels are populated according to a Boltzmann distribution and that a Boltzmann distribution is maintained during dissociation energy losses. To complete the CVDV model, a coupling factor identical to that of the CVD model is used. Like the CVD model, the coupling coefficient is used as a correction factor for the forward rate constant when the vibrational mode of motion is not in equilibrium.

The third model used in this study is the Coupled-Vibration-Dissociation-Vibration^{7,8} model with preferential dissociation (CVDV-Preferential). There are important differences between the CVD, CVDV, and the CVDV-Preferential models which arise from the changes in the initial assumptions. The first assumption states that the mechanism of vibrational relaxation progresses to a final Boltzmann distribution, based on an equilibrium temperature, and that it occurs through a series of Boltzmann distributions. The second assumption was that the

dissociation processes occur with equal probability from any vibrational level. The final assumption is that the vibrational energy system is modeled as a series of simple harmonic oscillators cutoff prior to dissociation.

In order to present a more physically correct representation of a physical system, it would be natural to change any assumptions that are based on ideal situations. Therefore, the second and third assumptions have been changed for the CVDV-Preferential model. The resulting model is comprised of the following expressions

$$\begin{aligned} \frac{\partial \epsilon_v}{\partial t} = & \frac{\epsilon_{v_\infty} - \epsilon_v}{\tau} + \frac{[\bar{E}(T, T_v) - \epsilon_v]}{n} \left[\frac{\partial n}{\partial t} \right]_f \\ & - \frac{[\bar{G}(T) - \epsilon_v]}{n} \left[\frac{\partial n}{\partial t} \right]_r \end{aligned} \quad (4)$$

$$\phi = \frac{k_f(T, T_v)}{k_\infty(T)} = \frac{Q(T)Q(T_m)}{Q(T_v)Q(-U)} \quad (5)$$

The individual terms depicted in equation (4) are defined in the same manner as for the CVDV model except that a greater probability of dissociation has been assigned to the higher vibrational energy levels. This preferential treatment and control of the higher energy levels is accomplished in the following manner. First, the Morse vibrational energy curve is replaced by an anharmonic

oscillator and mathematically represented by the following Taylor series expansion⁹

$$E_v = \omega_e(\epsilon_v + 1/2) - \omega_e x_e(\epsilon_v + 1/2)^2 + \omega_e y_e(\epsilon_v + 1/2)^3 - \omega_e z_e(\epsilon_v + 1/2)^4 \quad (6)$$

The expansion coefficients in equation (6) have been determined experimentally and are listed in Reference 9 for most monatomic, ionic, and diatomic species of air. The final step in preferential control is the selection of a characteristic probability temperature, which is located in the denominator of equation (5). A typical value of this temperature is assumed to be approximately one third to one sixth of the ideal dissociation temperature⁸. It should be noted that as the characteristic probability temperature approaches infinity, an equal probability of dissociation is assigned to each energy level. This condition corresponds to the previous assumptions of the CVD and CVDV models.

The final vibrational coupling model used in this study is the two-temperature chemical-kinetic model (TT_v) developed by Park¹⁰. The TT_v model assumes that the rotational temperature is characterized by, or in equilibrium with, the translational temperature and that the electronic and electron temperatures are in equilibrium with the vibrational temperature. The total rate of change of vibrational energy, as described by Park, is the summation of four individual

components. These components include the contributions of heavy particle and electron collisions as well as the removal of vibrational energy due to dissociation. Park also assumed a change in the functional dependency of the forward reaction rates. Since the TT_v model is based on two temperatures, the forward reaction rates with a general collision partner were based on a geometric average temperature

$$T_a = \sqrt{TT_v} \quad (7)$$

After several unsuccessful attempts of trying to incorporate the original Park model into the present computer algorithm, a Park-Like model was created for this study. These unsuccessful attempts were characterized by an inability to conserve mass at points in the flowfield, which appeared to be related to the use of T_a in the the forward reaction rates. The resulting Park-Like model contained the first term of the TT_v vibrational rate expression with its limited collisional cross section and diffusion modification, plus the second and third terms of the CVDV model in accordance with the description in Reference 10. The geometric average temperature located in the forward rates was replaced by the CVDV coupling factor. This substitution gave an equivalent effect similar to the average geometric temperature by slowing the forward dissociation processes. The Park-Like model yielded the following expressions

$$\frac{\partial \epsilon_v}{\partial t} = \frac{\epsilon_{v_\infty} - \epsilon_v}{\tau_L} \left| \frac{T - T_v}{T_s - T_{vs}} \right|^{(S-1)} + (CVDV)_2 + (CVDV)_3 \quad (8)$$

$$S = 3.5 e^{(-5000/T_s)} \quad (9)$$

$$\left\{ \begin{array}{l} \tau_L = \tau + \tau_c \\ \tau_c = 1 / (nC\sigma_v) \\ \sigma_v = 10^{-17} (50000/T)^2 \end{array} \right\} \quad (10)$$

Equation (8), which in the present study has only been applied to nitrogen, represents the vibrational rate equation of the CVD model, but with a correction factor on the first term to compensate for the non-linear behavior of the vibrational energy near the shock front at extreme temperatures. This phenomenon resembles a diffusion process; therefore, the coefficient is sometimes referred to as a diffusion factor. Equation (9) is referred to as a bridging formula since it connects the diffusion correction to the linear behavior of vibrational energy as originally derived by Landau-Teller¹¹. Similarly, the set of expressions depicted in equation (10) are corrections to the vibrational relaxation time¹². These expressions arise from the fact that the correlation formulas presented in Reference 12 do not limit, at high temperatures, the collisional cross section for vibrational relaxation process. Therefore, a limiting collisional cross section term is introduced in order to

ensure reasonable vibrational relaxation times.

The determination of accurate experimental data for vibrational relaxation times as a function of temperature is a crucial part of vibrational coupling model evaluation. Since some theories contain gross approximations, which have led different investigators to reach opposing conclusions, it became necessary to compare existing vibrational relaxation data and to make an appropriate selection. In order to select a reliable set of vibrational relaxation data, a comparison has been made between the experimental data of Blackman¹³, Millikan and White¹², and the data listed in Reference 2. The data in Reference 2 used the functional form of equation (11), which is the standard representation of vibrational relaxation time as predicted by classical Landau-Teller theory,

$$\tau = \frac{\alpha T e^{\beta (\sigma T^{-1/3})}}{p \left[1 - e^{(-\theta_v/T)} \right]} \quad (11)$$

while that in References 12 and 13 were of the form

$$p\tau = \alpha T e^{\beta (\sigma T^{-1/3})} \quad (12)$$

These experimental time correlations for nitrogen vibrational relaxation have been plotted using the standard Landau-Teller representation of $\log(p\tau)$ versus $T^{-1/3}$ and are

presented in Figure 2. Both the vibrational relaxation data of Reference 2 and Blackman display non-linear behavior over the entire temperature range due to their functional forms, while the Millikan and White experimental data shows a linear behavior due to the fact that β in equation (12) is assumed to be zero. As can be seen, the Millikan and White data has a greater rate of change with temperatures than the data of References 2 and 13. Nevertheless, after reviewing these correlations and experiments¹⁴, it was felt that the Millikan and White representation of vibrational data was the most accurate; therefore, it was subsequently used for this comparative study.

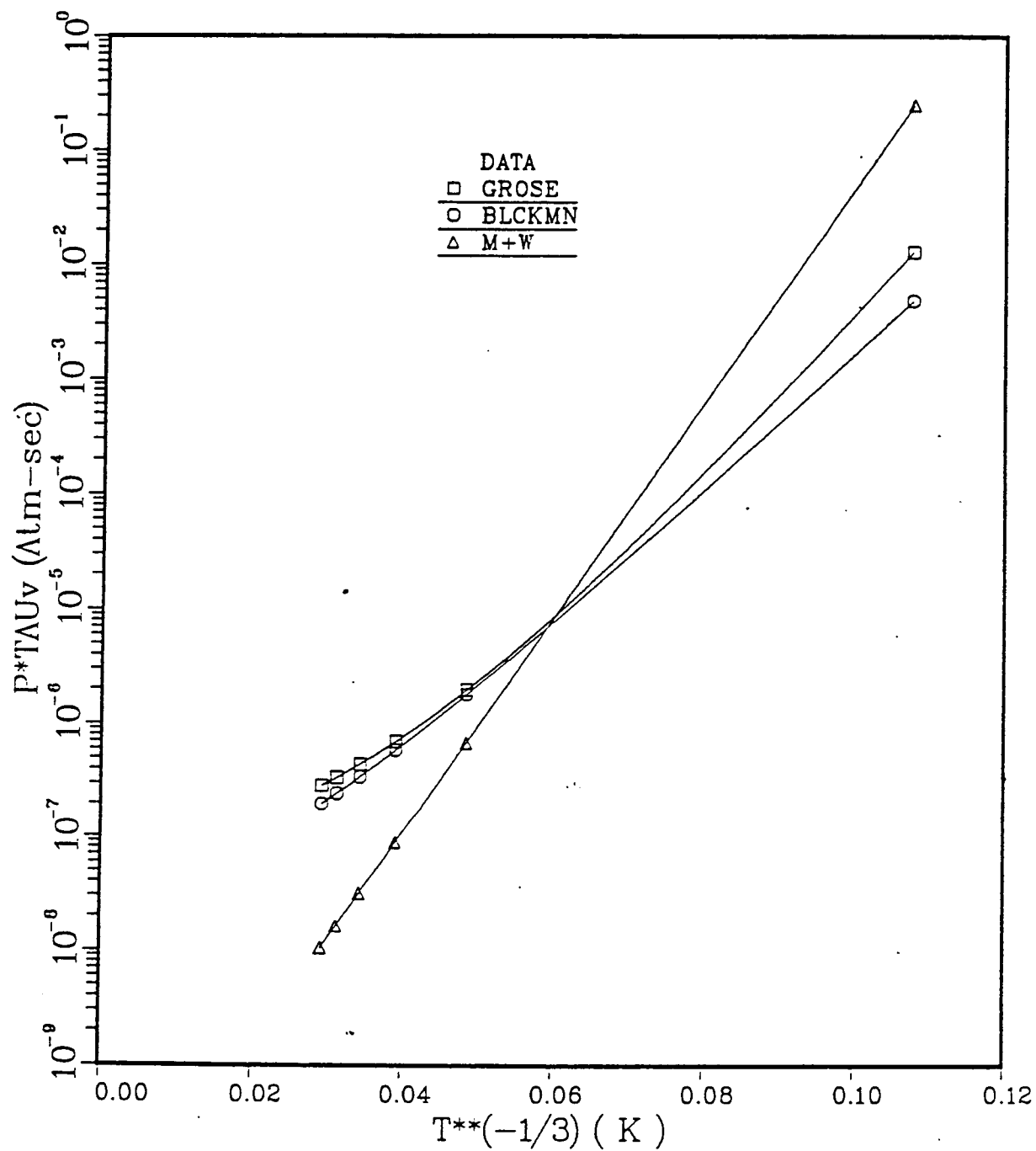
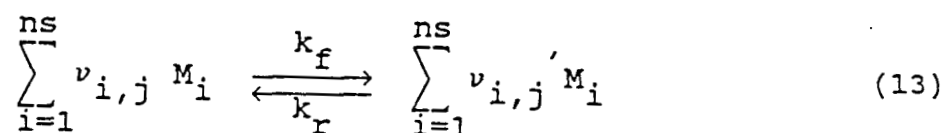


FIGURE 2. VIBRATIONAL RELAXATION DATA FOR NITROGEN

CHEMICAL KINETICS

Many mathematical representations can be used to describe a system of chemical reactions within a flowfield; however, such descriptions must reflect a coupling of the chemical reactions and species to the changing flowfield quantities. The nonequilibrium chemistry formulation, used in Reference 4, treated the forward and backward reactions as separate forward reactions during computation and required the specification of both the forward and backward reaction rate coefficients. For a given reaction, chemical rate data is not always available for both directions; and therefore, it became necessary to modify the present program so that one rate could be specified and the other could be determined based on the equilibrium coefficient of the reaction. Since the chemical formulation presented in Reference 7 had this capability of dual selection, its algorithm was incorporated into the present research program.

A physical system undergoing change, due to collisional exchanges, can be represented or described by a set of chemical reactions and species. These collisional exchanges can be expressed mathematically by the following expression,



where $i = 1, 2, \dots, n_s$ indicates the number of species and $j = 1, 2, \dots, n_r$ indicates the number of reactions within a given system. The left and right sides of equation (13) represent the reactants and the products respectively. The stoichiometric coefficients (ν) indicate the number of molecules of species M for both the products and the reactants.

For a condition of equilibrium to exist, the number of depleting collisions must equal the number of incoming collisions within a defined system. The collision rates, as represented by a specific reaction, can be determined by the forward and backward reaction rate coefficients which are related through an equilibrium constant. The term "constant" is frequently used because reaction chemistry is often studied experimentally at a fixed translational temperature.

The equilibrium constant, with its subsequent reaction rate expressions, is usually represented mathematically as

$$k_{eq} = \frac{k_{f\infty}}{k_{b\infty}} \quad (14)$$

$$k_{(f,b)\infty} = A T^B e^{(-E/T)} \quad (15)$$

In turn, the equilibrium constant can be expressed in terms of partial pressures through the change in free energy of the given chemical reaction; and this redefinition of the

equilibrium constant leads to

$$k_{p_j} = e^{(-\Delta F_j^\circ / RT)} \quad (16)$$

$$k_{eq_j} = k_{p_j} (RT)^{(\nu_{i,j}' - \nu_{i,j})} \quad (17)$$

Equations (16) and (17) are general forms of the law of mass action for mixtures of a thermally perfect gas. The thermodynamic state variable F° , known as the Helmholtz free energy, is normally defined as the difference between the total and irreversible energies of a macroscopic system. Using the statistical thermodynamic definition of free energy, the chemical system can also be expressed in the form of chemical potentials summed over all species for a specific reaction. Therefore, the free energy can be represented as

$$\frac{\Delta F_j^\circ}{RT} = \sum_{i=1}^{ns} (\nu_{i,j}' - \nu_{i,j}) \left[\frac{\mu_i^\circ}{T} \right] \quad (18)$$

where

$$\begin{aligned} \frac{\mu_i}{T} = & - \ln Q_{trans} - \ln Q_{rot} - \ln Q_{vib} - \ln Q_{elec} \\ & + \ln N_o + h_i^\circ \end{aligned} \quad (19)$$

Equation (19) represents the chemical potentials of a

Equation (19) represents the chemical potentials of a microscopic system in the form of factored partition functions for each molecular mode of motion. The standard heat of formation term in equation (19) is included in order to reference each species component to a common energy state.

The final phase in describing a physical gas system is the representation of the chemical production rates, which are customarily expressed in terms of the concentrations of individual species. This mathematical representation is based on collision theory which is in the form of a general expression derived from the law of mass action. This representation also requires that the rate of production of a given species be directly proportional to the product of the concentrations of the reacting species raised to the power equal to its stoichiometric coefficient. Therefore, the law of mass action is applied to equation (13) and results in the following rate expression for general production

$$\frac{d\left(\frac{x}{\rho}\right)}{dt} = \frac{(v_{i,j}' - v_{i,j})}{\rho} \left[k_{f\infty} \phi \prod_{n=1}^{ns} (x)^{v_{n,j}} - k_{b\infty} \prod_{n=1}^{ns} (x)^{v_{n,j}'} \right] \quad (20)$$

where the notation is that of Reference 7. Since the solution method that was used for this study is a function of a given x shock coordinate and stream function, equation (20) must be altered accordingly to

$$\begin{aligned}
 \left(\frac{d(c_i)}{dx} \right)_{\psi} &= \frac{(v_{i,j}' - v_{i,j})}{\rho u} \left[k_{f_{\infty}} \prod_{n=1}^{ns} \left(\frac{\rho c_i}{\mu_{m_i}} \right)^{v_{n,j}'} - k_{b_{\infty}} \prod_{n=1}^{ns} \left(\frac{\rho c_i}{\mu_{m_i}} \right)^{v_{n,j}} \right] \\
 &\cdot \left[\rho \sum_{i=1}^{ns} \frac{c_i a_{i,j}}{\mu_{m_i}} \right]^{(I_{\max} + 1)} \quad (21)
 \end{aligned}$$

SOLUTION METHOD AND ANALYSIS

The solution technique that was chosen for this study is an inverse method which was first derived by Maslen³ and later modified by Grose⁴ to include chemical nonequilibrium processes. The Maslen method is formulated in a shock-oriented or curvilinear coordinate system which is shown in Figure 3. The governing equations for continuity, momentum, and energy are then transformed into the Von Mises plane. The governing equations in the non-transformed coordinates are

Continuity:

$$\frac{\partial(\rho u r)}{\partial x} + \frac{\partial(\rho v \lambda r)}{\partial y} = 0 \quad (22)$$

Momentum:

$$u \frac{\partial u}{\partial x} + \lambda v \frac{\partial u}{\partial y} - \frac{uv}{R_c} + \frac{1}{\rho} \frac{\partial p}{\partial x} = 0 \quad (23)$$

$$u \frac{\partial v}{\partial x} + \lambda v \frac{\partial v}{\partial y} + \frac{u^2}{R_c} + \frac{\lambda}{\rho} \frac{\partial p}{\partial y} = 0 \quad (24)$$

Energy:

$$h + \frac{u^2}{2} + \frac{v^2}{2} = \text{constant} \quad (25)$$

Rate eqn:

$$u \frac{\partial c_n}{\partial x} + \lambda v \frac{\partial c_n}{\partial y} = \lambda \omega_n(p, \rho, c_1, c_2, \dots, c_n) \quad (26)$$

Equation of state:

$$h = h(p, \rho, c_1, c_2, \dots, c_n) \quad (27)$$

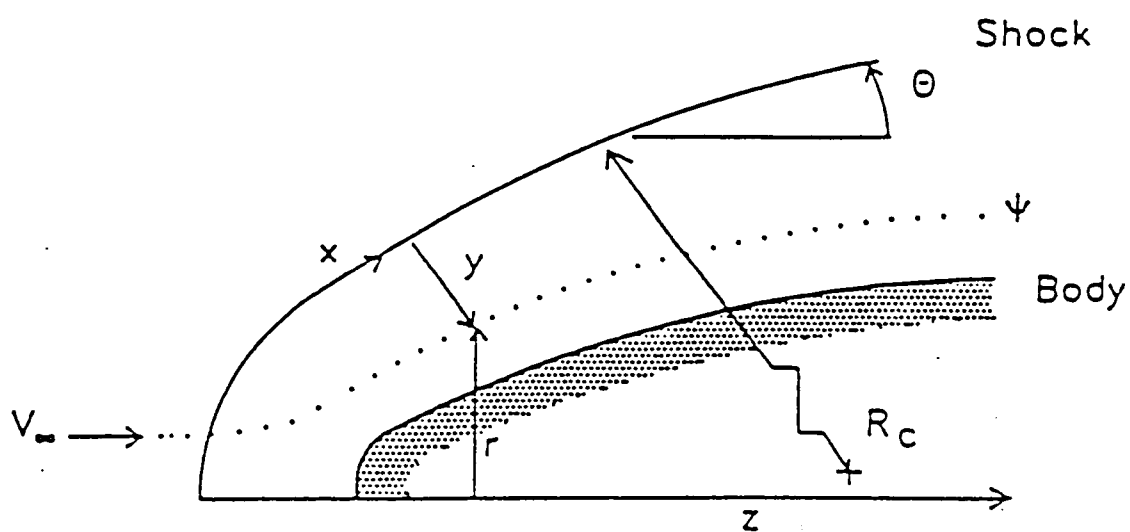


FIGURE 3. SHOCK-ORIENTED COORDINATE SYSTEM

The Von Mises transformation allows the governing equations to be redefined so that the independent variables become x , distance along the shock, and the stream function ψ . The actual transformation equations can be derived from the partial derivative of the stream function as defined from the continuity equation. As given in Reference 4, these equations are

$$\frac{\partial \psi}{\partial x} = \rho v r \lambda \quad (28)$$

$$\frac{\partial \psi}{\partial y} = -\rho u r \quad (29)$$

$$\left(\frac{\partial}{\partial x} \right)_{\psi} = \frac{\partial}{\partial x} + \frac{\lambda v}{u} \frac{\partial}{\partial y} \quad (30)$$

By using equation (30), the transformed expressions for the momentum and rate equations become

$$\left(\frac{\partial u}{\partial x} \right)_{\psi} - \frac{v}{R_c} + \frac{\lambda v r}{u} \left(\frac{\partial p}{\partial x} \right)_x + \frac{1}{\rho u} \left(\frac{\partial p}{\partial x} \right)_{\psi} = 0 \quad (31)$$

$$\left(\frac{\partial v}{\partial x} \right)_{\psi} + \frac{u}{R_c} - \lambda r \left(\frac{\partial p}{\partial x} \right)_x = 0 \quad (32)$$

$$\left(\frac{\partial c_n}{\partial x} \right)_{\psi} = \frac{\lambda \omega_n}{u} \quad (33)$$

Again, by using the relationship in equation (30) and the transformed momentum equations, the energy equation can be obtained in the (x, ψ) plane. The final form can be obtained

by differentiating equation (25) and combining it with equations (23) and (24) which yields

$$\left(\frac{\partial h}{\partial x}\right)_\psi = \frac{1}{\rho} \left(\frac{\partial p}{\partial x}\right)_\psi \quad (34)$$

By utilizing the thin shock layer approximation, which assumes that the flow is almost parallel to the shock, a simplified version of the transformed lateral momentum equation can be obtained. Thus, equations (32) and (33) can be simplified by assuming that $\left(\frac{\partial v}{\partial x}\right)_\psi$ is small and that the scale factor λ is equal to one. The effect of these approximations is to uncouple the lateral momentum equation from the remaining equations so that an expression for pressure independent of the flowfield chemistry can be obtained. Equation (32) can then be integrated on a line normal to the shock to obtain the pressure as a function of the specified shock geometry and flow values immediately behind the shock. This leads to the following approximation first derived by Maslen

$$p(x, \psi) = p_s(x) + \frac{u_s(x)}{R_c(x) r_s(x)} \left[\psi - \psi_s(x) \right] \quad (35)$$

Notice that equation (35) is a function only of the shock geometry and shock jump chemistry. In a related study on shock jump chemistry performed by Greendyke¹⁵, oxygen, or oxygen and nitrogen were assumed to immediately dissociate to

equilibrium concentrations at the shock front, and the consequences of these assumptions on computational efficiency and flow properties were investigated.

From the requirement of energy conservation, the velocity at any point in the flowfield can subsequently be determined based on the parallel shock flow approximation of $(v^2 - v_s^2) \ll u^2$

$$u = \sqrt{2(h_o - h)} \quad (36)$$

As with any numerical method, the proper specification of boundary conditions is essential. Since the Maslen method is an initial value problem, the specification of all starting values at the shock front are determined by shock geometry, assumed shock jump chemistry, and freestream conditions. In addition, the stream function values associated with the shock and the body are

$$\psi_s = \frac{\rho_\infty V_\infty r_s^2}{2} \quad (37)$$

$$\psi_b = 0 \quad (38)$$

The final step in this method is the transformation from the curvilinear coordinates (x, ψ) back to the physical (z, r) space. By inspecting Figure 3, one can deduce three important transformations from trigonometry. These relations are

$$r = r_s(x) - y \cos \theta_s(x) \quad (39)$$

$$z = z_s(x) + y \sin \theta_s(x) \quad (40)$$

$$u_s(x) = \cos \theta_s(x) \quad (41)$$

where equation (41) is a reiteration of the parallel shock flow assumption. The final form of the physical space transformation equation is obtained by first inverting equation (29) and then integrating with respect to ψ . The resulting equation obtained after the substitution into equation (39) is

$$r = \left[r_s^2 - 2u_s \int_{\psi}^{\psi_s} \frac{d\psi}{\rho u} \right]^{1/2} \quad (42)$$

SOLUTION PROCEDURE

An inverse method is a technique in which an initial quantity, other than a body-type boundary condition, is specified. From an initial description of a shock shape using an analytical expression of the form

$$r_s = r_s(z) \quad (43)$$

The local shock angle, radius of curvature, and distance along the shock can be determined by the following expressions

$$\theta_s = \tan^{-1} \left[\frac{dr_s}{dz} \right] \quad (44)$$

$$R_c = \frac{\left[1 + \left(\frac{dr_s}{dz} \right)^2 \right]^{3/2}}{\left| \frac{d^2 r_s}{dz^2} \right|} \quad (45)$$

$$x = \int_0^{r_s} \frac{dr_s}{\sin \theta_s} \quad (46)$$

Initiation of the solution begins with the determination of these geometric shock values at all streamline points, which are located by user selected values of delta-x and maximum z.

In addition, preselected freestream values of temperature, pressure, velocity, and gas composition are used to compute the freestream density, molecular weight, and internal energy. The following mathematical expressions are used in conjunction with a constraint to calculate these freestream quantities

$$\rho_{\infty} = \frac{\mu_{m_{\infty}} P_{\infty}}{RT_{\infty}} \quad (47)$$

$$\mu_{m_{\infty}} = \left[\sum_{i=1}^{I_{\max}} \frac{c_i}{\mu_{m_i}} \right]^{-1} \quad (48)$$

$$\sum_{i=1}^{I_{\max}} c_i = 1 \quad (49)$$

$$e_{\infty} = \sum_{i=1}^{I_{\max}} c_i e_i \quad (50)$$

The next step within the computer algorithm is the determination of the thermodynamic state of the gas immediately behind the shock wave, which is sometimes referred to as the shock jump condition. This condition usually assumes an instantaneous equilibration of the translational, rotational, and electronic contributions to the internal energy of the gas at some translational temperature T_s . The vibrational contribution to the internal

energy can be treated as being in equilibrium with the translational mode or as being in vibrational nonequilibrium as modeled through a vibrational coupling model. In addition, the species concentrations across the shock are assumed to remain constant at their freestream values, which is referred to as frozen flow.

In order to calculate the translational temperature across the shock front, an iterative procedure is required to solve the oblique shock relations. These relations as expressed from the conservation of mass, momentum, and energy are

$$\Lambda(x) = \rho_{\infty} V_{\infty} \sin \theta_s \quad (51)$$

$$\Omega(x) = p_{\infty} + \rho_{\infty} V_{\infty}^2 \sin^2 \theta_s \quad (52)$$

$$\delta(x) = e_{\infty} + \frac{p_{\infty}}{\rho_{\infty}} + \frac{V_{\infty}^2 \sin^2 \theta_s}{2} \quad (53)$$

With an initial estimate of T_s , values of internal energy and temperature across the shock front can be calculated using equation (50) and the following relation

$$T_s = \frac{2\mu_{m_{\infty}}(\delta - e_s) \left[\Omega^2 - 2\Lambda^2(\delta - e_s) \right]^{1/2}}{R \left\{ \Omega + \left[\Omega^2 - 2\Lambda^2(\delta - e_s) \right]^{1/2} \right\}} \quad (54)$$

Equation (54) can then be iterated using a Newton's method

until subsequent values of T_s differ by less than a preselected tolerance. Upon convergence, the shock jump values of pressure, density, enthalpy, and velocity are evaluated by

$$p_s = \left[\Omega^2 - 2\Lambda^2(\delta - e_s) \right]^{1/2} \quad (55)$$

$$\rho_s = \frac{p_s \mu_{m_\infty}}{RT_s} \quad (56)$$

$$h_s = e_s + \frac{p_s}{\rho_s} \quad (57)$$

$$u_s = V_\infty \cos \theta_s \quad (58)$$

The next step in the solution procedure is the determination of the pressure and the pressure gradient for the entire flowfield. First, the pressure is evaluated in the (x, ψ) plane using equation (35). Then, the value of the pressure gradient, which is used primarily to evaluate the change in enthalpy along a given streamline, is computed through second order finite difference representations. This step, as well as the aforementioned steps, provide a set of initial conditions which are necessary for a stepwise evaluation of the rate of change associated with species concentrations, vibrational energy, and enthalpy along streamlines.

The next, and most important, part of the solution

algorithm is the evaluation of the partial derivatives along a given streamline, and the obtaining of the flowfield properties through numerical integration. Unfortunately, there is an undesirable characteristic associated with the governing equations which model chemically reacting flows. This characteristic corresponds directly to the so called "stiff behavior" of the chemical rate equations, which are extremely sensitive to step size. The physical reason for this behavior is the rapidity at which a set of chemical reaction rates can change during dissociation, recombination, and ionization processes.

Grose⁴ conducted a study to determine which numerical integration technique would minimize the unstable and stiff behavior associated with a chemically reacting system. For his analysis, he selected the fourth order Runge-Kutta, Adams-Moulton, and Treanor¹⁶ methods. The Treanor method was chosen because of its significant reductions in computational times, its ability to efficiently handle stiff systems, and automatically alter step size. This latter feature reduced the chance of divergence through a series of internal checks. As the flow approached chemical equilibrium, this method automatically increased the integration step size in order to increase the rate of convergence and reduce the necessary computational time. The Treanor integration formula is based upon approximating $\frac{dy}{dx} = f(x,y)$ and is comprised of the following expression,

$$\begin{aligned}
y = \tilde{h} \left\{ f_1 F_1 + \left[-3(f_1 + Py_1) + 2(f_2 + Py_2) + 2(f_3 + Py_3) \right. \right. \\
- (f_4 + Py_4) \left. \right] F_2 + 4 \left[(f_1 + Py_1) - (f_2 + Py_2) \right. \\
\left. \left. - (f_3 + Py_3) + (f_4 + Py_4) \right] F_3 \right\} \quad (59)
\end{aligned}$$

where \tilde{h} is the Δx interval of integration and the functions F_n and related constants are defined by

$$f_i = f(x_i, y_i) \quad (60)$$

$$\left[\begin{array}{ll} x_1 = x & , \quad y_1 = y \\ x_2 = x_1 + \tilde{h}/2 & , \quad y_2 = y_1 + (\tilde{h}/2)f_1 \\ x_3 = x_2 & , \quad y_3 = y_1 + (\tilde{h}/2)f_2 \\ x_4 = x_1 + \tilde{h} & , \quad y_4 = y_1 + (\tilde{h})f_3 \end{array} \right] \quad (61)$$

$$P = \frac{f_3 - f_2}{y_3 - y_2} \quad (62)$$

$$F_0 = e^{-P\tilde{h}} \quad (63)$$

$$F_n = \frac{F_{n-1} - \frac{1}{(n-1)!}}{-P\tilde{h}} \quad (64)$$

It should be noted that as P approaches zero, equation (59) reduces to the fourth order Runge-Kutta method.

The integration step is then accomplished in the

following manner. First, an initial integration is performed in accordance with a preselected Δx value. This integration step corresponds to a new point on the chosen streamline at which the new streamline point is at the new x value and along a y coordinate perpendicular to the shock front. The values of pressure and pressure gradient are then determined for that point, and at the midpoint of the interval, through the use of interpolation. The value of internal energy is then evaluated, using an initial estimate for translational temperature and interpolated pressure, at the initial point of the interval by a Newton iterative procedure. After convergence, the partial derivatives associated with species concentrations, vibrational energy, and enthalpy are evaluated at the start of the integration interval. This procedure, starting with the pressure interpolation, is then repeated twice at the middle and once at the end of the interval, as part of a fourth order Runge-Kutta technique, in order to obtain the necessary coefficients used in the Treanor numerical integration algorithm¹⁶. These coefficients reach their final values after the prescribed integration step has passed through a series of internal checks. If for any reason the step has produced an invalid result, such as a negative species concentration, the step size is reduced and the procedure is then repeated. After a successful completion of a step, the flowfield values are updated at the new point, whereby the vibrational temperature is computed for all diatomic species

based on their current values of vibrational energy. The overall procedure is repeated until each streamline has been integrated to an x-value which corresponds to the preselected maximum z-value.

While the above paragraph describes the method of solution for the entire flowfield, a special problem occurs when the chemical and thermodynamic quantities are initially evaluated for the body streamline ($\psi = 0$). The problem takes place since x and ψ both approach zero near the z-axis and cause the equations to become ill-behaved. The body-streamline integration is accomplished by dividing the initial delta-x step, within the stagnation region, into a set of subintervals. Each subinterval streamline is integrated in accordance with the flowfield algorithm until the x value is equal to the initial delta-x value. At which point, all the thermodynamic and chemical quantities are stored. Final evaluation for the body point is accomplished by the extrapolation of all stored quantities to the $\psi = 0$ body point. The normal algorithm then proceeds from there using the extrapolated quantities for initial values.

The final step in the solution procedure is the determination of the physical space coordinates for the body and flowfield streamlines. These coordinates are obtained by integrating equation (42) from the given streamline to the shock using interpolated delta-psi and $(1/\rho u)$ values along lines of constant x. The flow chart for the complete solution algorithm is depicted in Figure 4.

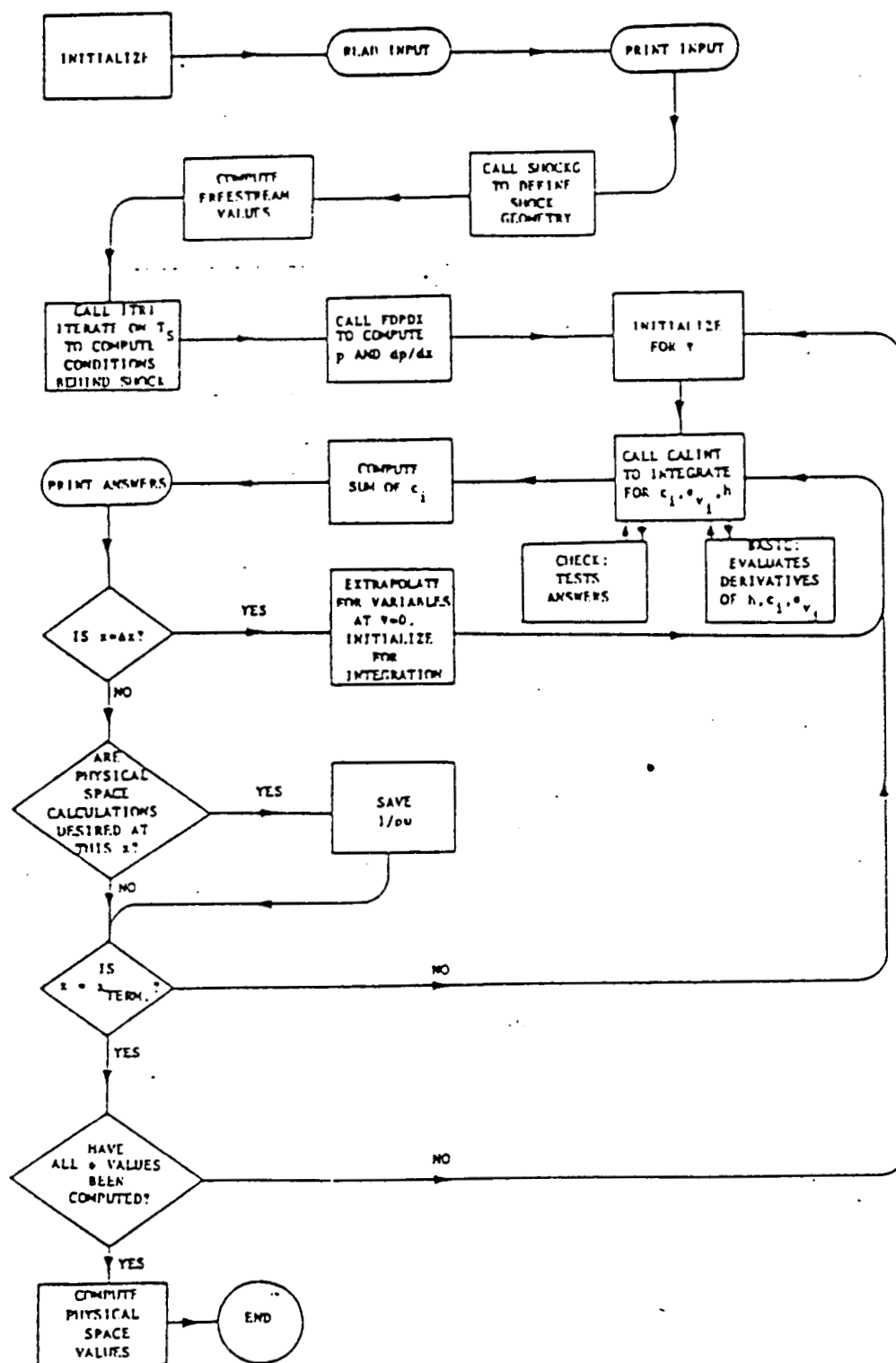


FIGURE 4. FLOW CHART OF SOLUTION ALGORITHM
(Adapted from Reference 4)

INTERNAL ENERGY MODEL

Since the AOTV flowfield is assumed to be composed of perfect gases, which implies that the intermolecular forces surrounding a particle are negligible, the molecular activity within the system can be represented as a series of elastic particle collisions. Further, the assumption that the collisional activity at sufficiently high temperatures distributes the particles over a wide range of permissible energy states, is the basic foundation of a reacting gas model. This distribution is referred to as a Boltzmann distribution, and it allows the energy states within a given system to be represented in terms of partition functions which depict how the particles are distributed or partitioned among the various energy states. It is convenient to express these partition functions in terms of energy levels instead of energy states. An energy level is defined as consisting of all energy states having identical values of energy. A level containing more than one state is defined as a degenerate level. The internal energy within a given set of levels can also be subdivided into its modes of motion using partition functions. These modes can include contributions from translation, rotation, vibration, and electronic excitation. Thus, the internal energy can be represented as a summation of each molecular mode over a range of energy states and the internal energy for the i^{th} species is

$$e_i = e_{t_i} + e_{r_i} + e_{v_i} + e_{e_i} + \Delta h_i^\circ / \mu_{m_i} \quad (65)$$

where the last term, the standard heat of formation, is included in order to reference all species to a common state.

The internal energy per unit mass, as derived from statistical thermodynamics¹⁷, can be expressed as

$$e = \frac{RT^2}{\mu_m} \left[\frac{\partial (\ln Q)}{\partial T} \right] \quad (66)$$

where the partition functions for the translational, rotational, vibrational, and electronic modes are

$$Q_{\text{trans.}} = V_m \left[\frac{2\pi mkT}{\hat{h}^2} \right]^{3/2} \quad (67)$$

$$Q_{\text{rot.}} = \frac{8\pi^2 IkT}{\sigma_m \hat{h}^2} \quad (68)$$

$$Q_{\text{vib.}} = \frac{1}{1 - e^{(-\theta_v/T_v)}} \quad (69)$$

$$Q_{\text{elec.}} = \sum_{\ell=1}^L g_{i,\ell} e^{(-\epsilon_{i,\ell}/T)} \quad (70)$$

where k is the Boltzmann constant, m is the mass of the molecule, V_m is the specific volume and \hat{h} is Planck's constant. In addition, I represents the molecular moment of inertia, σ_m is a symmetry factor, and g_ℓ is the degeneracy

factor of energy level l .

The substitution of these molecular partition functions into equation (66) yields the following expressions

$$e_{t_i} = \frac{3}{2} \left[\frac{RT}{\mu_{m_i}} \right] \quad (71)$$

$$e_{r_i} = f_i \left[\frac{RT}{\mu_{m_i}} \right] \quad (72)$$

$$e_{v_i} = \frac{f_i R \epsilon_i}{\mu_{m_i} \left[e^{(\epsilon_i / T_{v_i})} - 1 \right]} \quad (73)$$

$$e_{e_i} = \frac{R}{\mu_{m_i}} \frac{\sum_{l=1}^{L_i} g_{i,l} \epsilon_{i,l} e^{(-\epsilon_{i,l}/T)}}{\sum_{l=1}^{L_i} g_{i,l} e^{(-\epsilon_{i,l}/T)}} \quad (74)$$

The total internal energy at a given point is then obtained by substituting equation (65) into equation (50).

RESULTS AND DISCUSSION

For this discussion, the basic AOTV mission profile has been divided into three areas of interest, which are referred to as the entry, max-Q, and exit trajectory points. The values of freestream pressure, temperature, and velocity for each point are displayed in Table 1. A flowfield analysis at each point has been made with each vibration-dissociation coupling model using three specific reaction chemistry sets. This analysis, which consisted of forty five separate computations, has been followed by a comparison study of all vibration-dissociation coupling models. The study has been further discretized into an analysis of the three chemical reaction rate sets at each trajectory point for individual vibration-dissociation coupling models. The chemical reaction systems and their respective rate coefficients are depicted in Tables 2, 3, and 4.

The comparative studies have been conducted in order to determine which combination of species, reaction rates, and vibration coupling models is most representative of the AOTV flowfield as well as having the additional advantage of computational efficiency. For completeness, vibrational equilibrium results have been included for discussion purposes even though they were not part of the actual comparative study. In addition, all shock jump conditions have been based on the assumption that the species

Table 1. AOTV Trajectory Points

Point	Press. (dynes/cm ²)	Temp. (K)	Vel. (cm/sec)
Entry	10.35	180.65	1000000
Max Q	15.715	197.101	891500
Exit	10.35	180.65	771000

Table 2. Reaction Rate Set 1 (RR1)

Reaction	A	B	E	Dir.
$N_2 + M \rightleftharpoons 2N + M$	5.0E+19	-1.5	0.0	k_b
$O_2 + M \rightleftharpoons 2O + M$	1.19E+21	-1.5	59380	k_f
$NO + M \rightleftharpoons N + O + M$	5.18E+21	-1.5	75490	k_f
$N + O_2 \rightleftharpoons NO + O$	1.0E+12	0.5	3120	k_f
$O + N_2 \rightleftharpoons NO + N$	5.0E+13	0.0	38016	k_f
$N + O \rightleftharpoons NO^+ + e^-$	1.8E+21	-1.5	0.0	k_b

Table 3. Reaction Rate Set 2 (RR2)

Reaction	A	B	E	Dir.
All of RR1				
$N + N \rightleftharpoons N_2^+ + e^-$	1.40E+13	0.0	67800	k_f
$O + M \rightleftharpoons O^+ + e^- + M$	2.77E+12	0.5	157800	k_f
$N + N^+ \rightleftharpoons 2N^+ + e^-$	2.34E+11	0.5	120000	k_f
$N + N \rightleftharpoons N^+ + e^- + N$	2.34E+11	0.5	120000	k_f
$N + e^- \rightleftharpoons N^+ + 2e^-$	4.16E+13	0.5	120000	k_f

Table 4. Reaction Rate Set 3 (RR3)

Reaction	A	B	E	Dir.
All of RR1				
$N + N \rightleftharpoons N_2^+ + e^-$	1.40E+13	0.0	67800	k_f
$O + M \rightleftharpoons O^+ + e^- + M$	2.77E+12	0.5	157800	k_f
$N + N^+ \rightleftharpoons 2N^+ + e^-$	2.34E+11	0.5	120000	k_f
$N + N \rightleftharpoons N^+ + e^- + N$	2.34E+11	0.5	120000	k_f
$N + e^- \rightleftharpoons N^+ + 2e^-$	1.10E+32	-3.14	169000	k_f

concentrations remained constant across the shock. This assumption is referred to as "frozen flow", and these frozen concentrations were used as starting conditions for each streamline calculation.

Since each of the forty five analyses involved 23 streamlines, only a small portion of the results can be represented. Specifically, the present discussion will concern only the streamline which crossed the shock front 1.5 cm. above the centerline. In addition, the mathematical function of

$$r_s = \left\{ 2 r z_s - B_s z_s^2 \right\}^{1/2} \quad (75)$$

where $r = 230$ and $B_s = -4$, has been used to represent the shock shape in order to obtain a 60 degree blunt cone as seen in Figure 1(a).

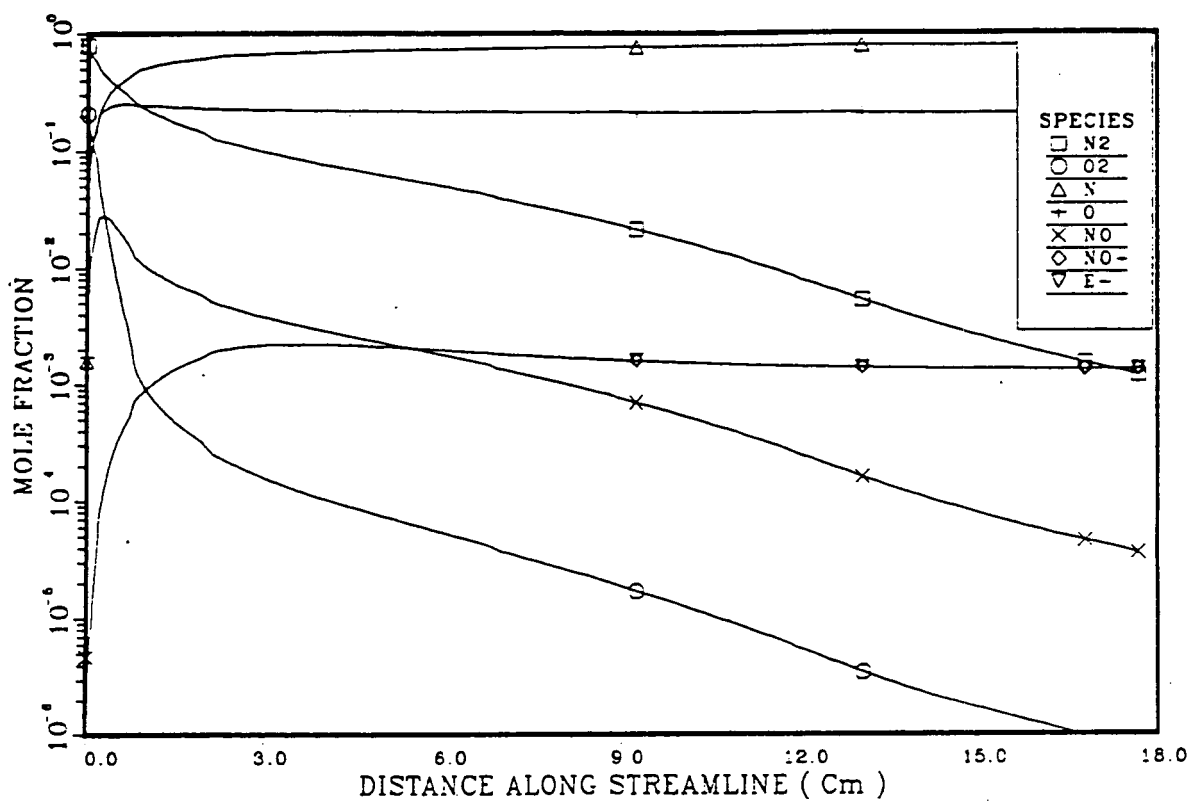
Reaction Rate Set 1.

Reaction rate set 1 (RR1)⁷ has been used as the primary air model for describing the processes of dissociation within the AOTV flowfield. The (RR1) set contains seven species and six reactions with (M) being the symbol for an arbitrary collision partner. For (RR1), the fourth reaction ($O + N_2 \rightleftharpoons NO + N$) has been considered to be the primary dissociating reaction for this mission profile. In addition, the ionization effects have been taken into account with the

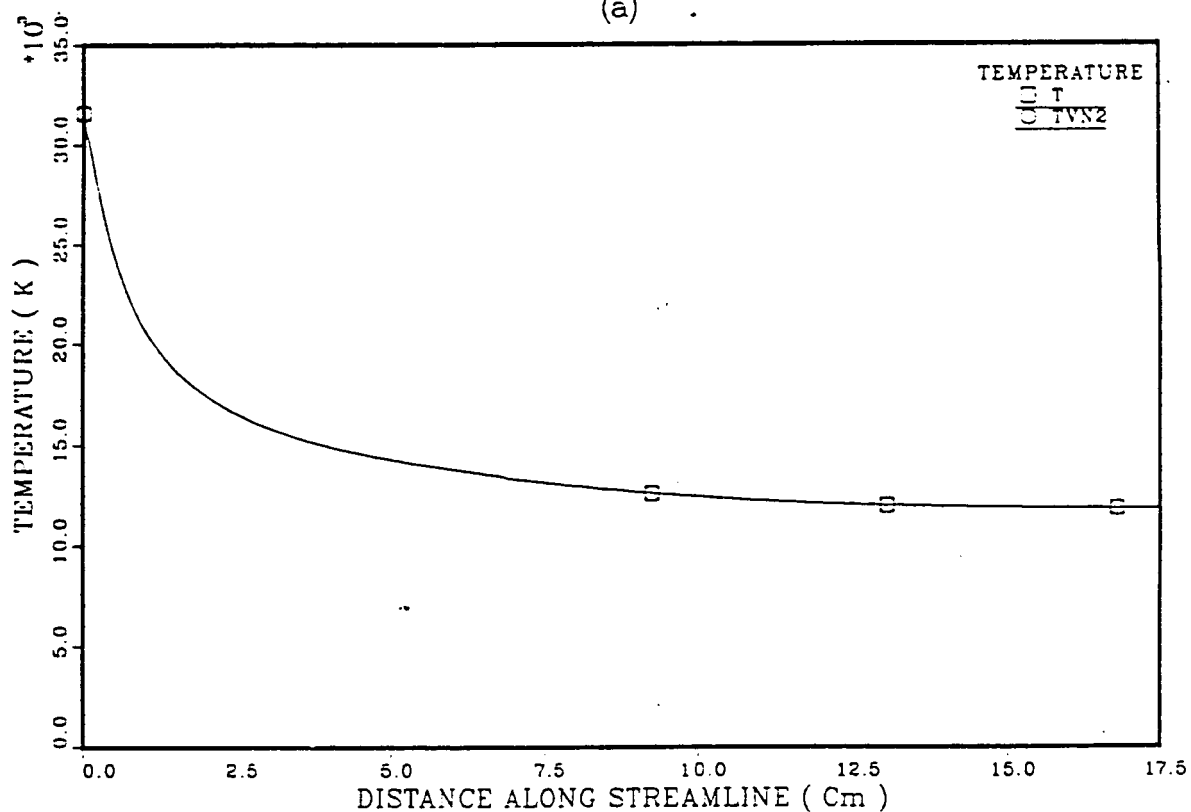
inclusion of the sixth reaction ($N + O \leftrightarrow NO^+ + e^-$). Since it is known from previous studies that ionization cannot be modeled entirely by one simple reaction for this type of flowfield, the main theme behind this reaction rate set has been to compare its results with the larger sets of reactions in order to possibly justify a savings in computational effort. The reaction rate coefficients on Tables 2-4 are of the Arrhenius form $[k = A T^B e^{(-E/T)}]$, and it should be noted that many of these coefficients were determined at temperatures lower than those which would exist in the AOTV shock layer. Nevertheless, it has been assumed that these coefficients are still reasonably accurate when used at higher temperatures.

The concentration and temperature profiles for vibrational equilibrium, CVD, CVDV, CVDV-Preferential, and the Park-Like models are displayed in Figures 5-9 at the entry point of 10 km/sec. In addition, comparison plots of the various vibrational coupling models for single flowfield variables are displayed in Figures 10-13.

For the vibrational equilibrium case, the vibrational energy component, used to compute the translational temperature across the shock, has been assumed to be equal to the translational energy mode. This assumption produces an instantaneous equilibration of translational and vibrational temperatures which in turn produces a higher rate of dissociation for various species due to an increase in vibrational energy. This trend can be seen in the N_2 , O_2 , and

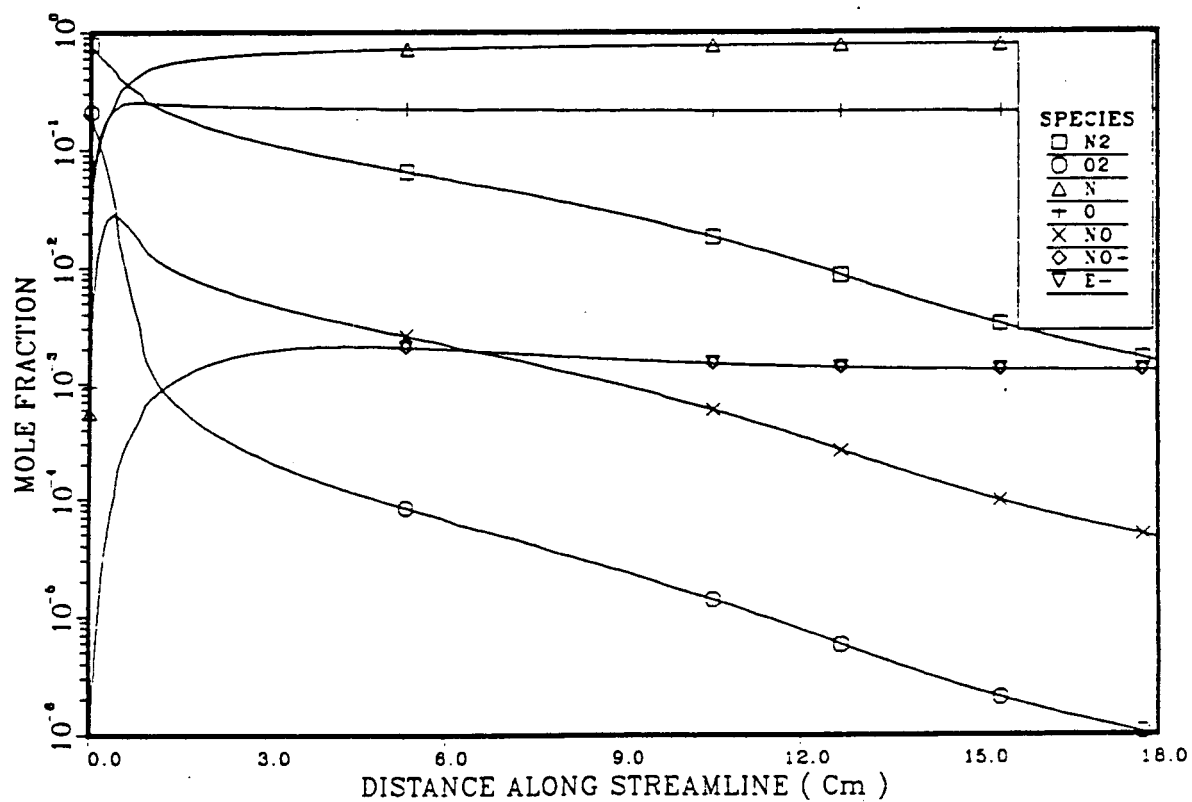


(a)

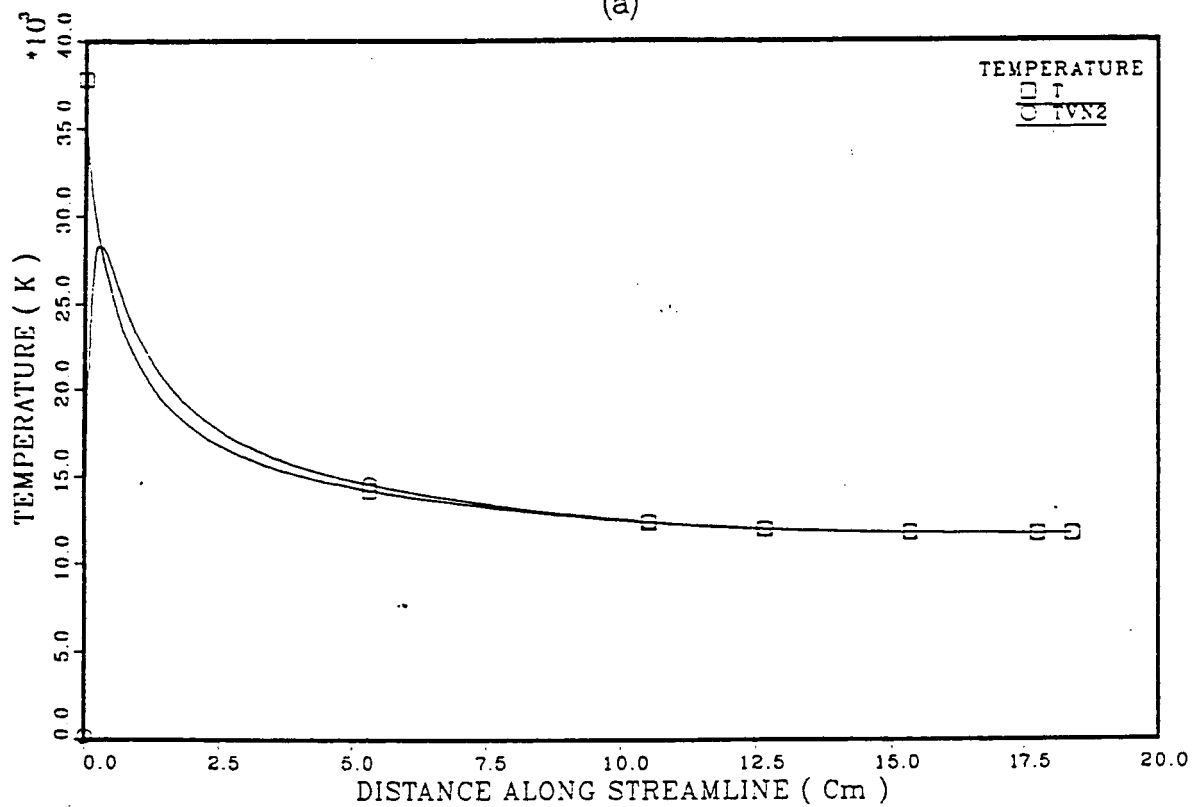


(b)

FIGURE 5. VEQ MODEL AT V=10 Km/s, RR1

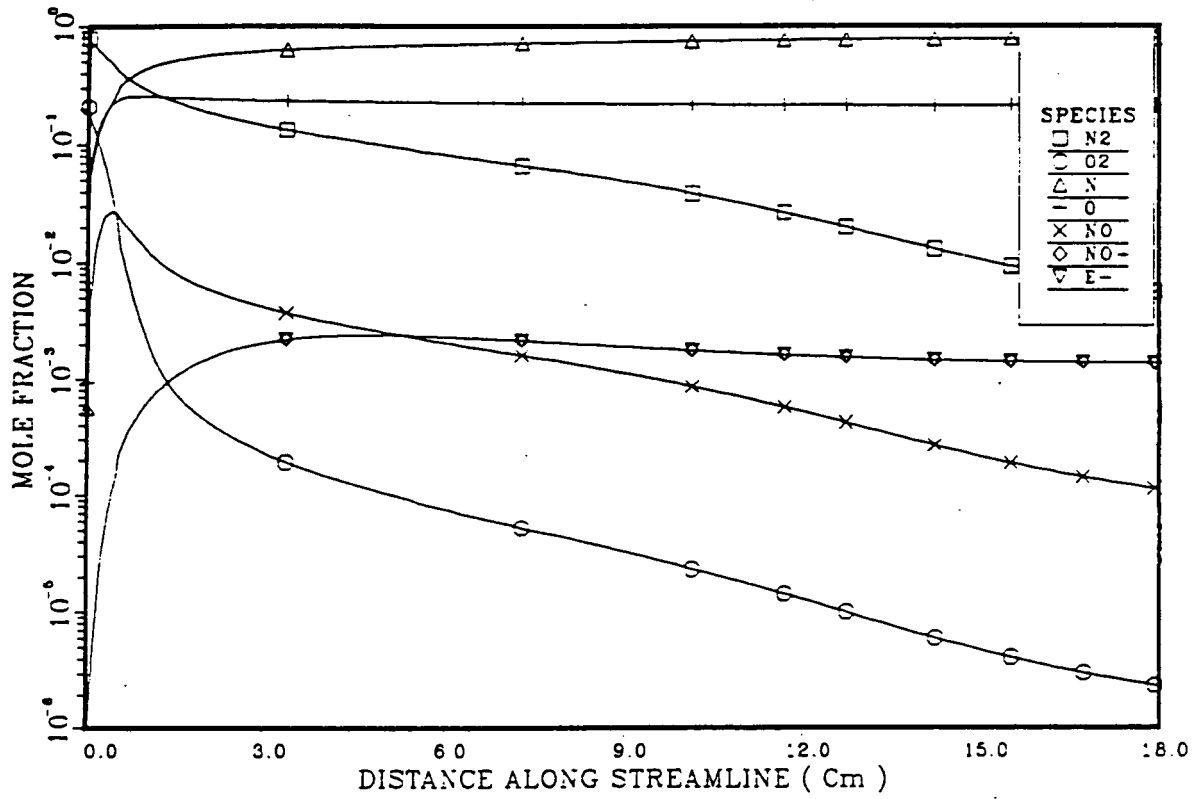


(a)

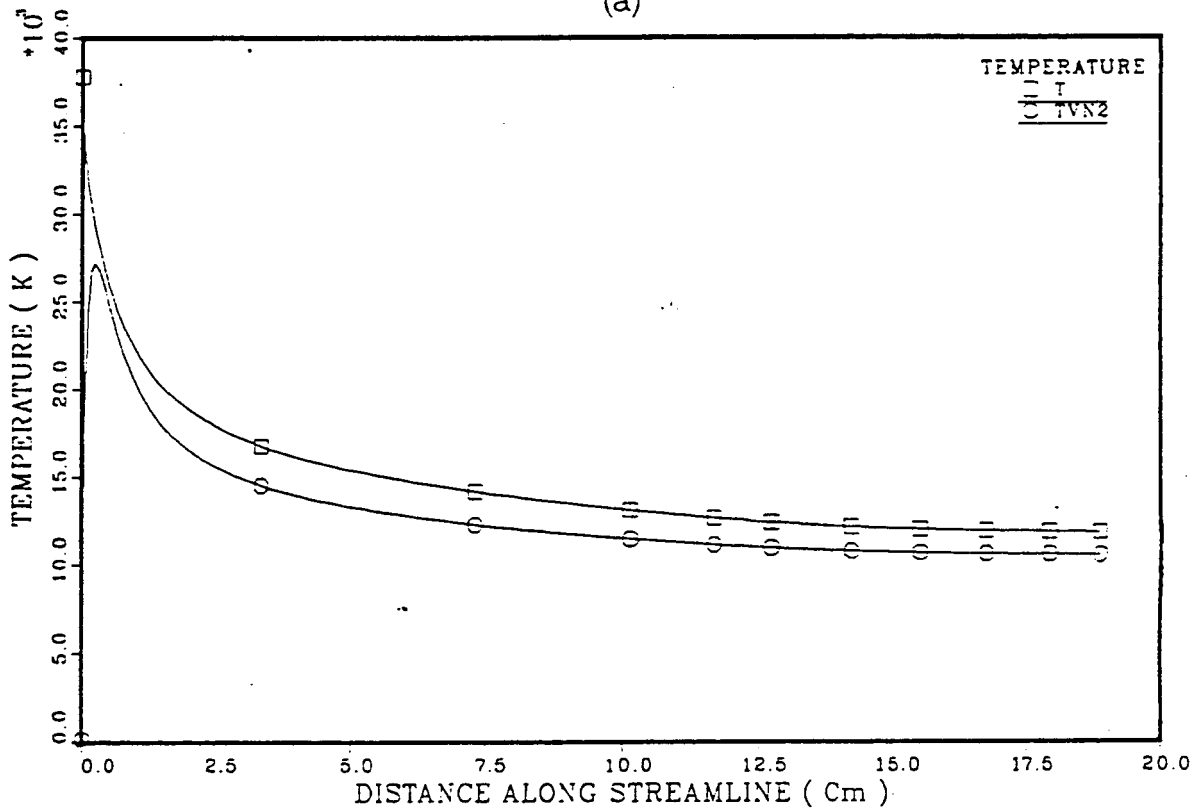


(b)

FIGURE 6. CVD MODEL AT $V=10$ Km/s, RR1

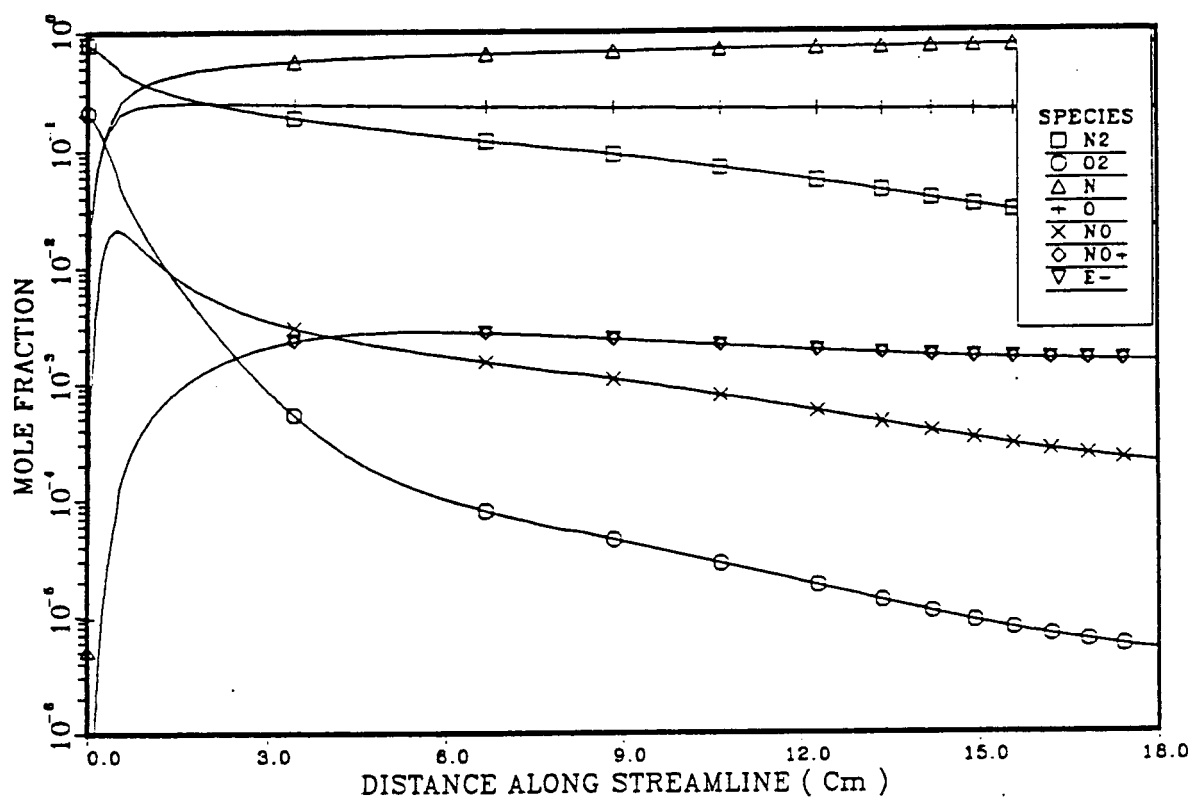


(a)

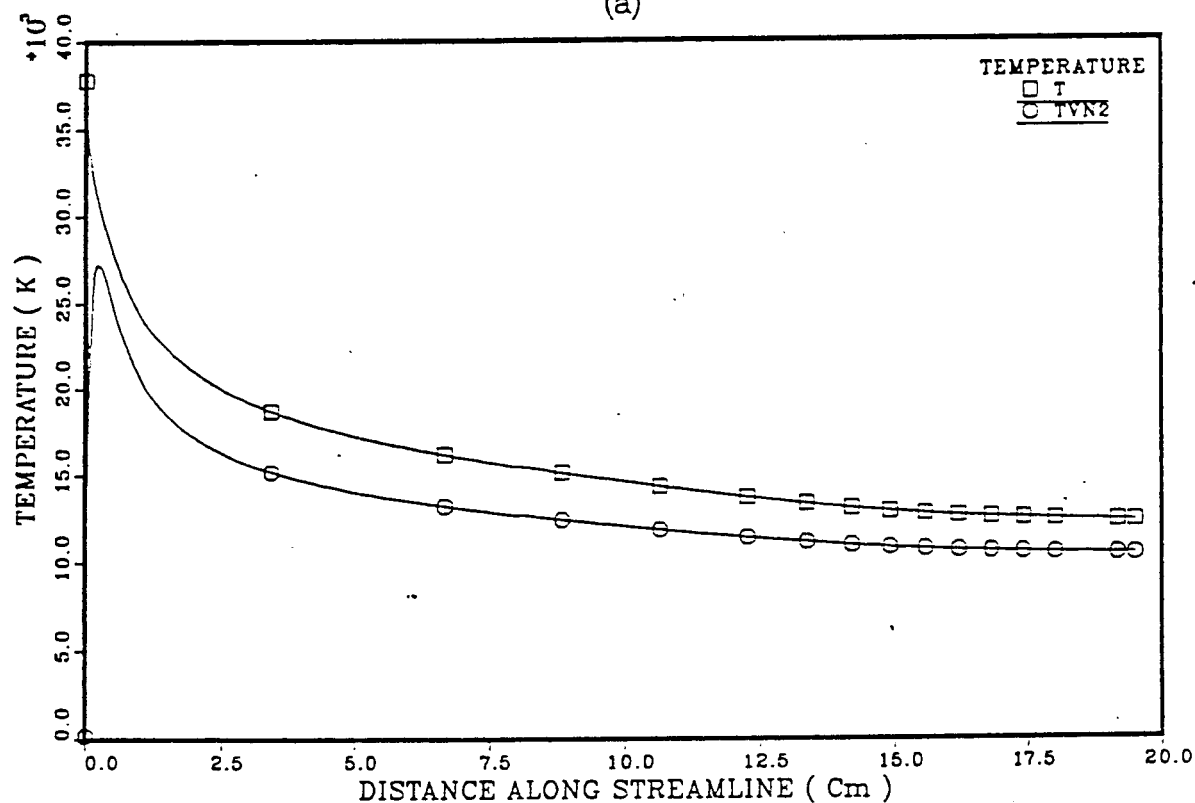


(b)

FIGURE 7. CVDV MODEL AT V=10 Km/s, RR1

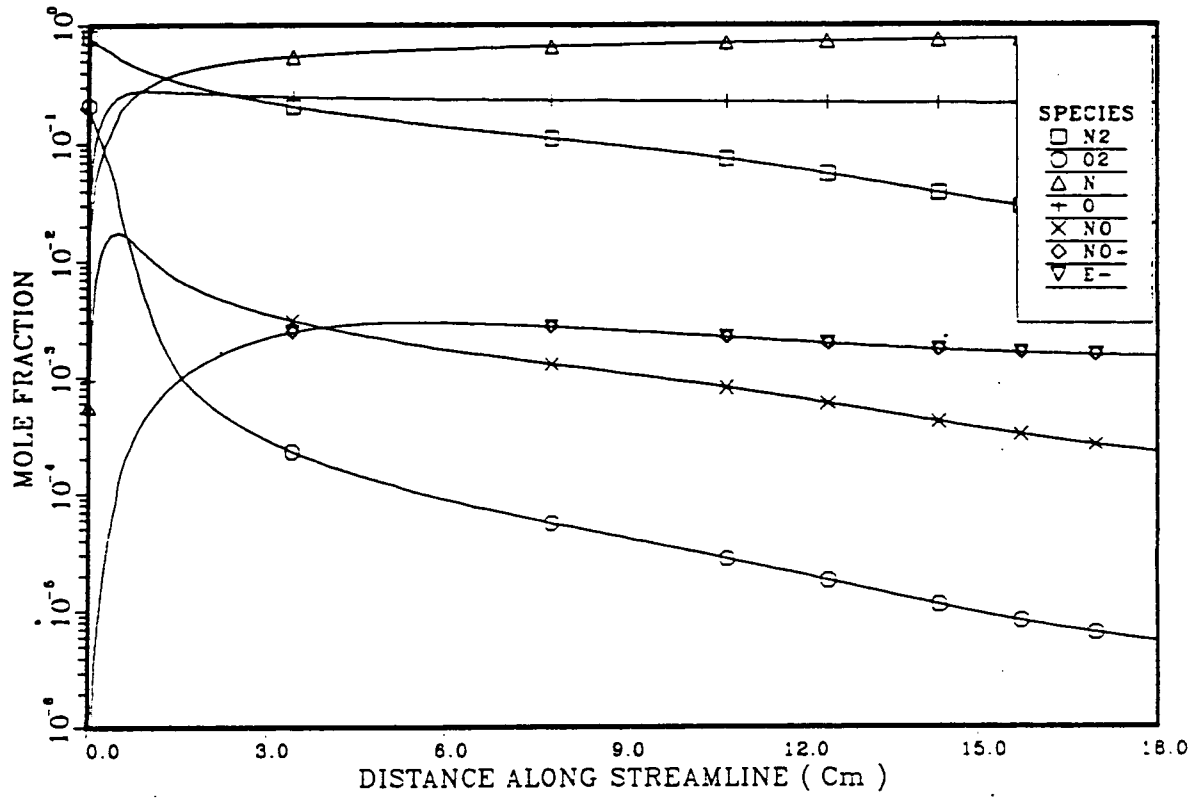


(a)

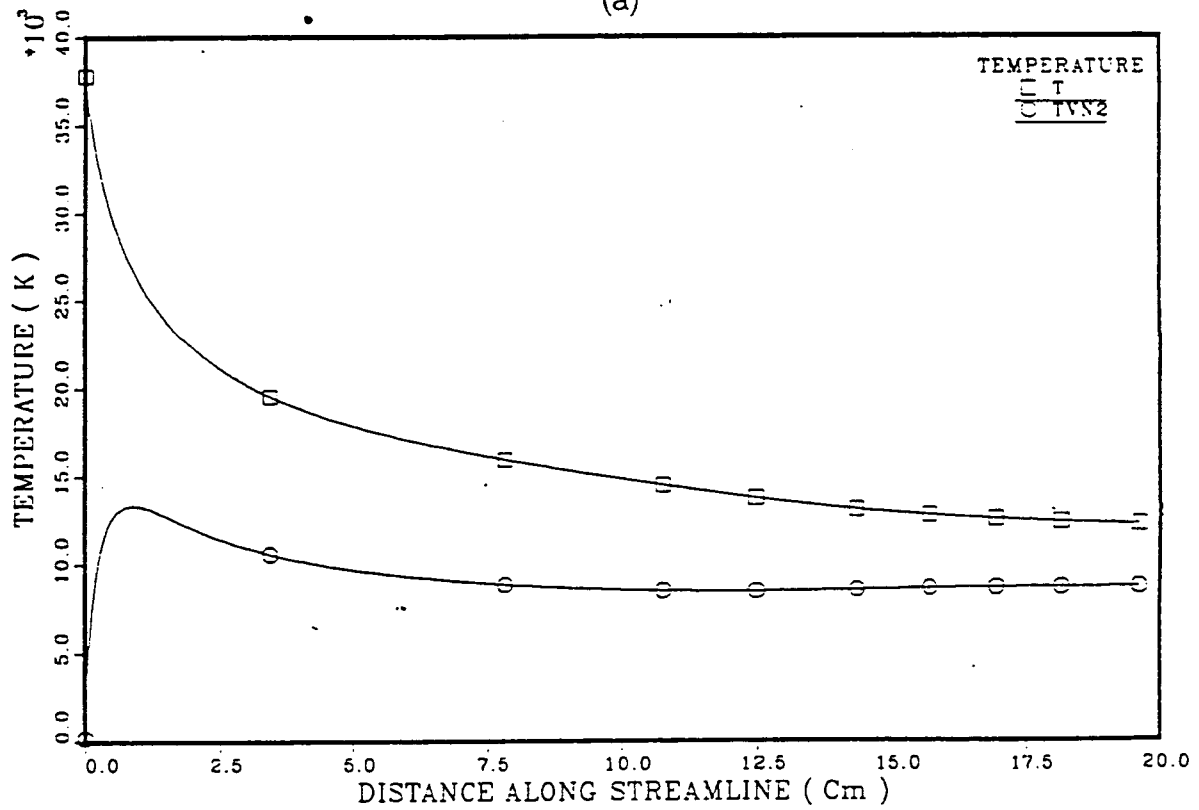


(b)

FIGURE 8. CVDV-P MODEL AT V=10 Km/s, RR1

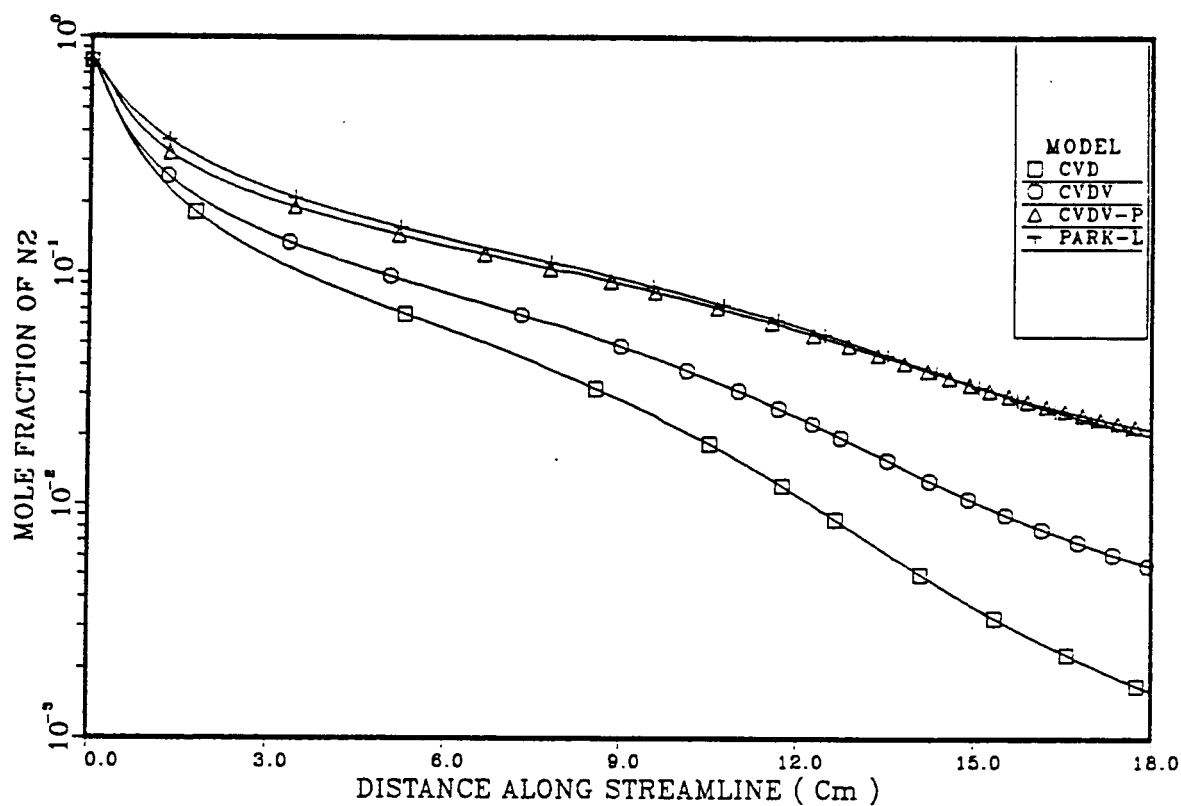


(a)

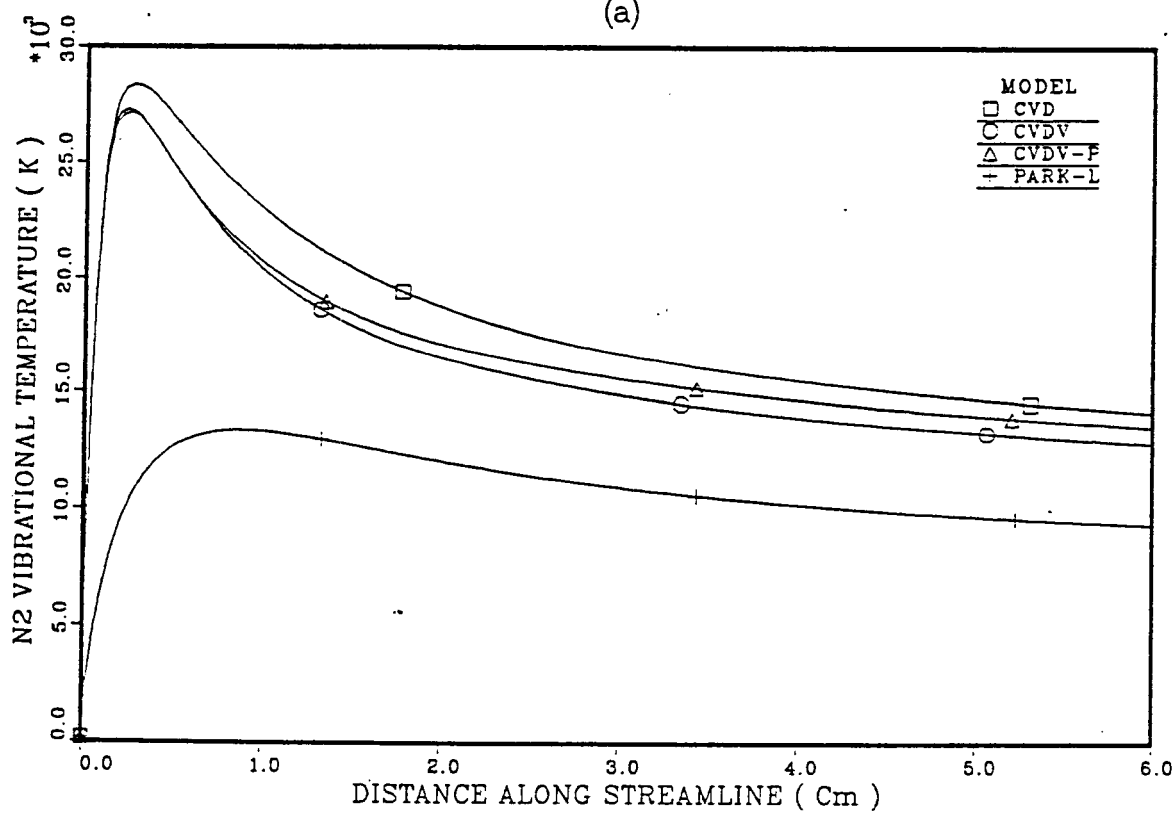


(b)

FIGURE 9. PARK-L MODEL AT $V=10$ Km/s, RR1

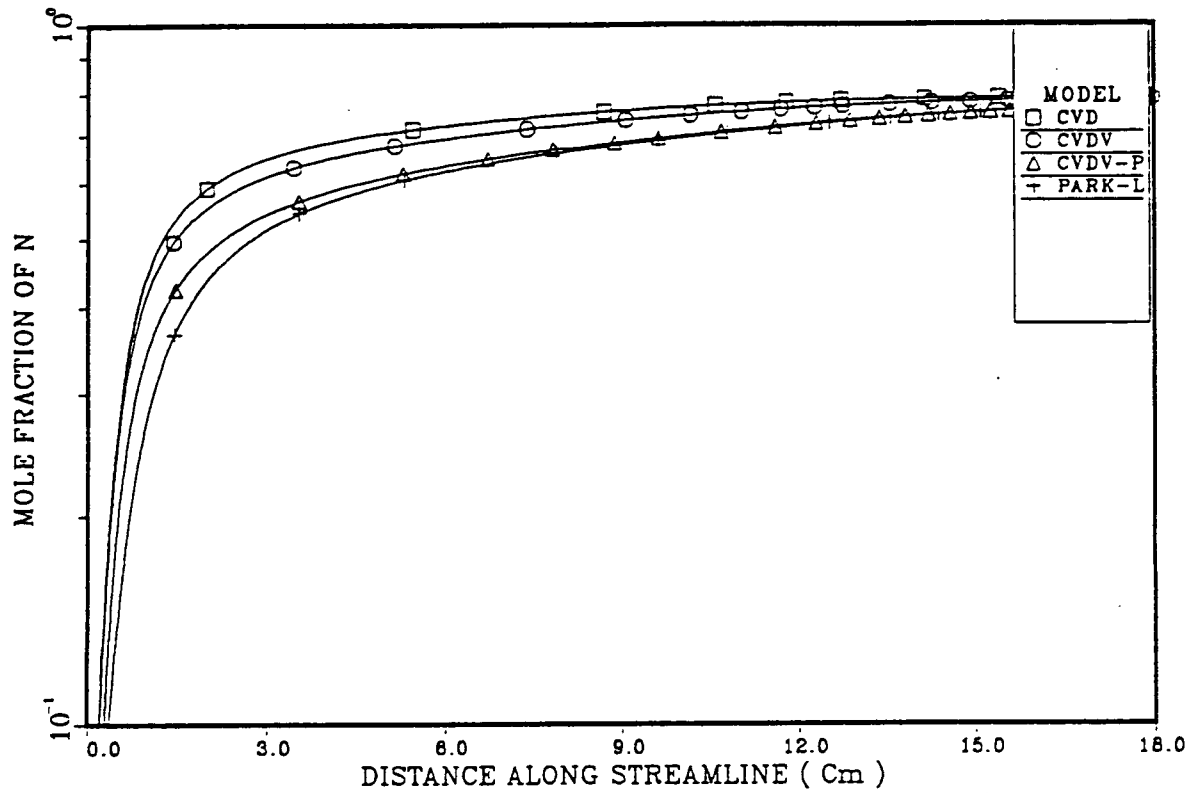


(a)

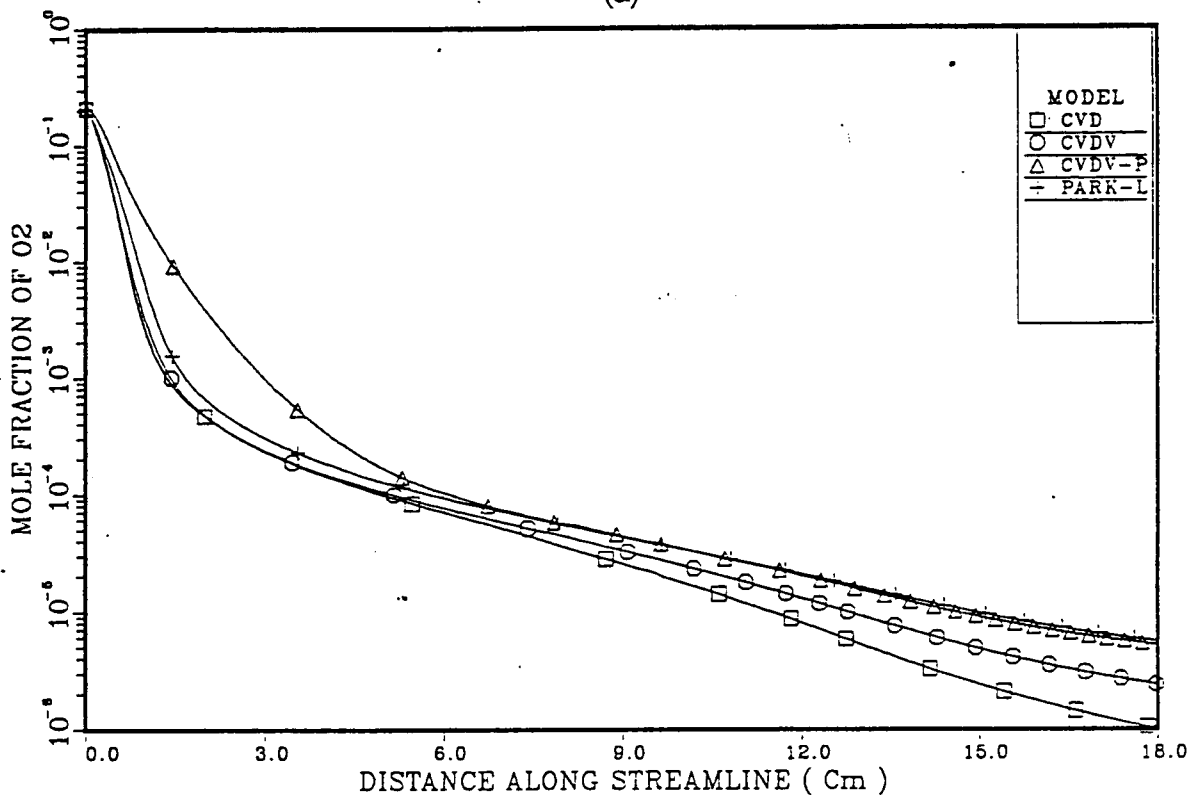


(b)

FIGURE 10. TVN2 AND N2 PROFILES AT V=10 Km/s, RR1

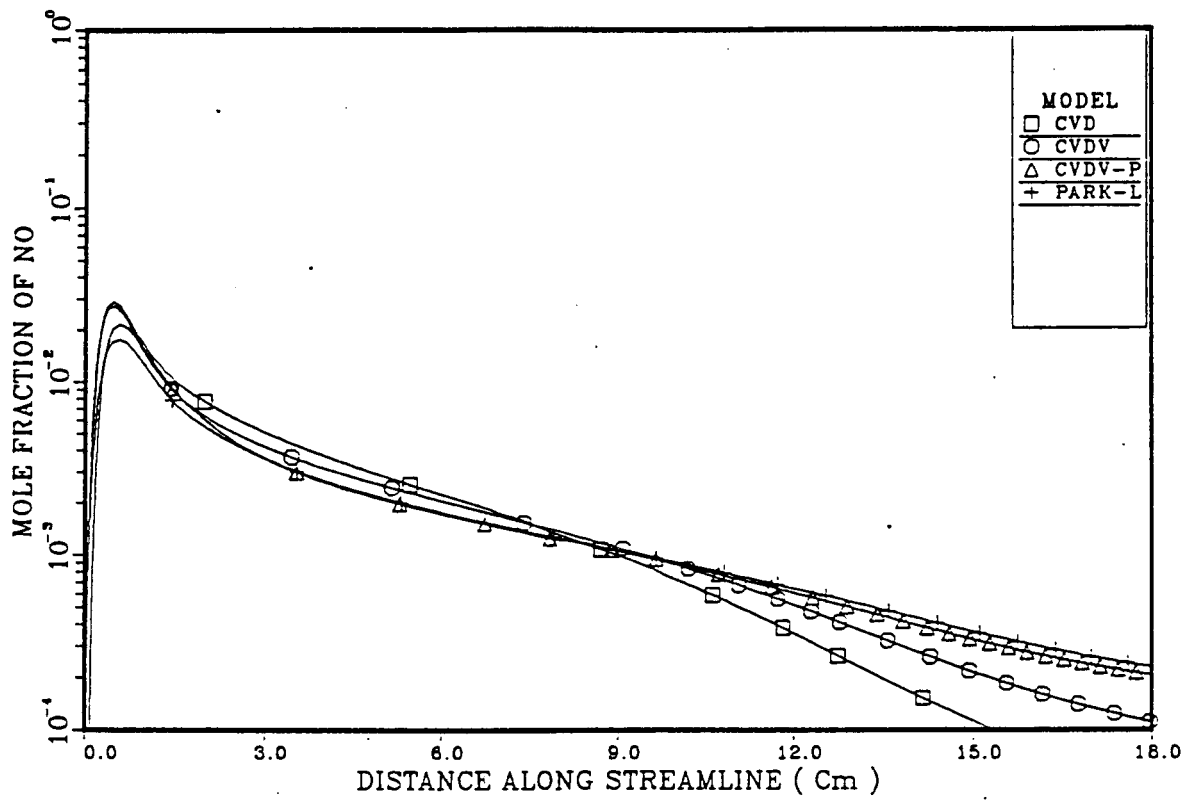


(a)

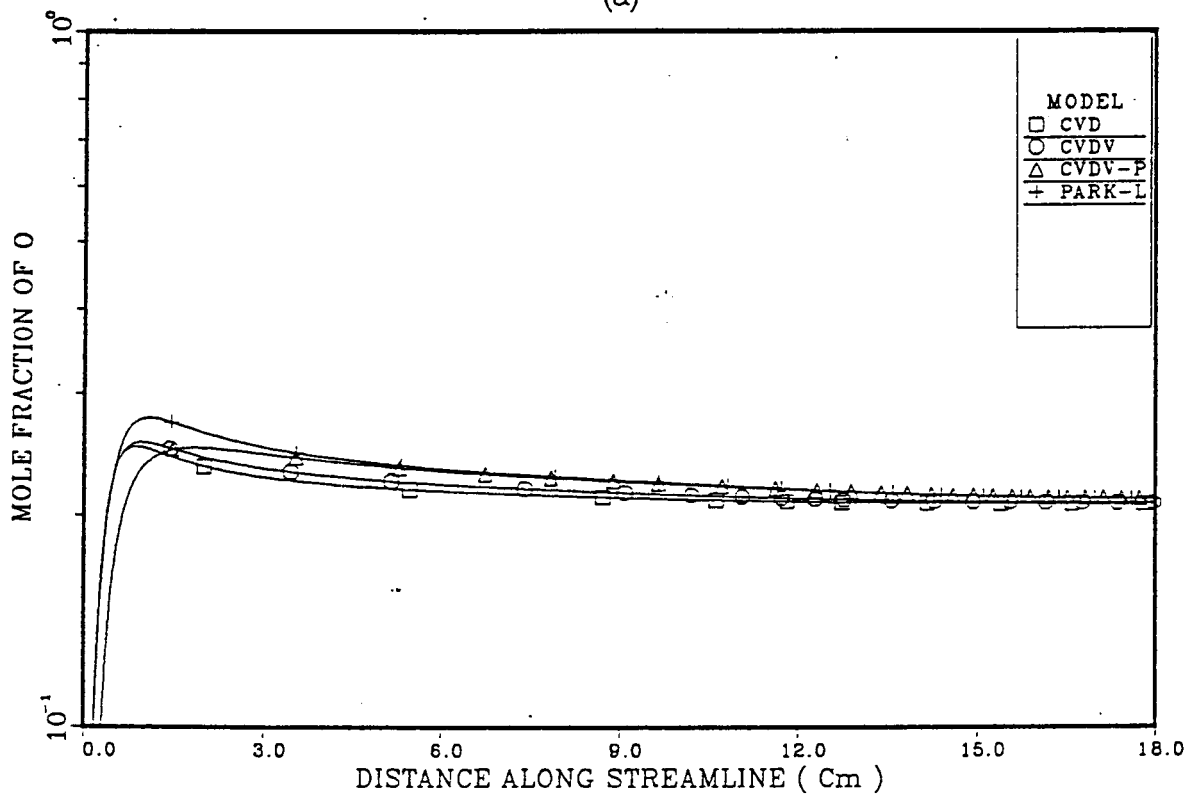


(b)

FIGURE 11. O2 AND N PROFILES AT V=10 Km/s, RR1

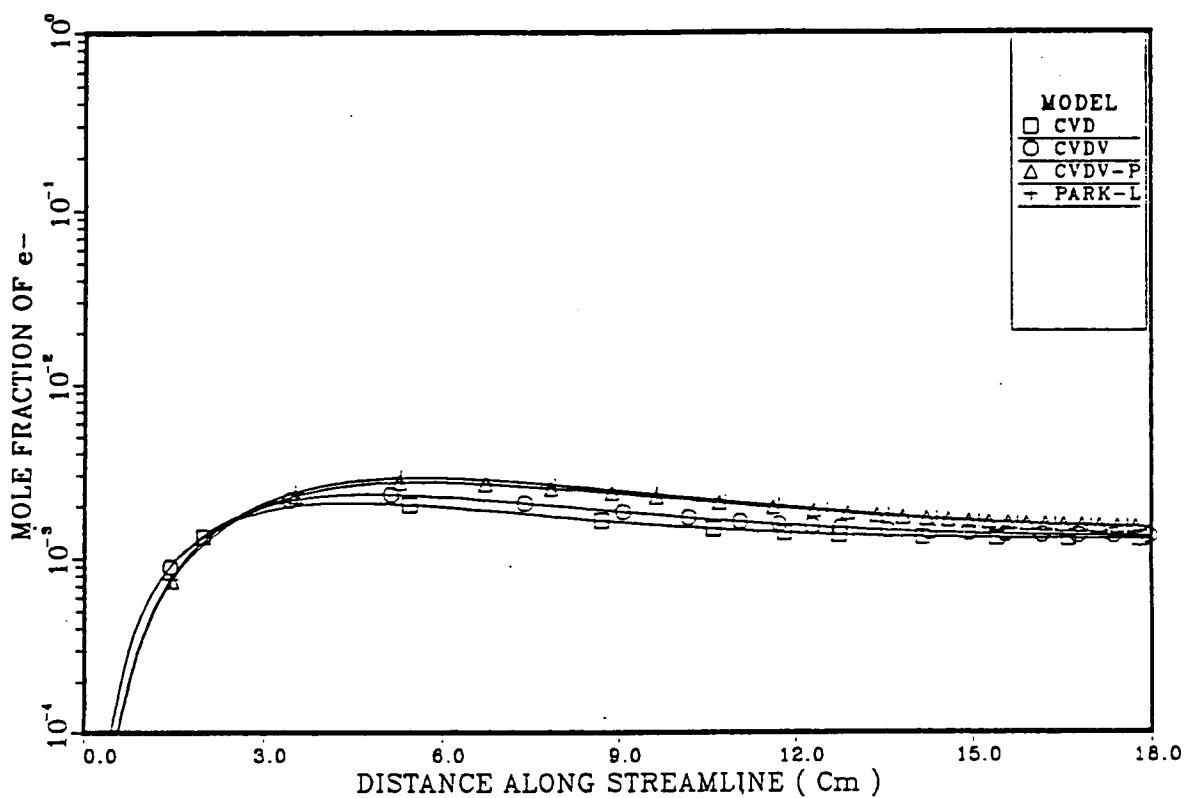


(a)

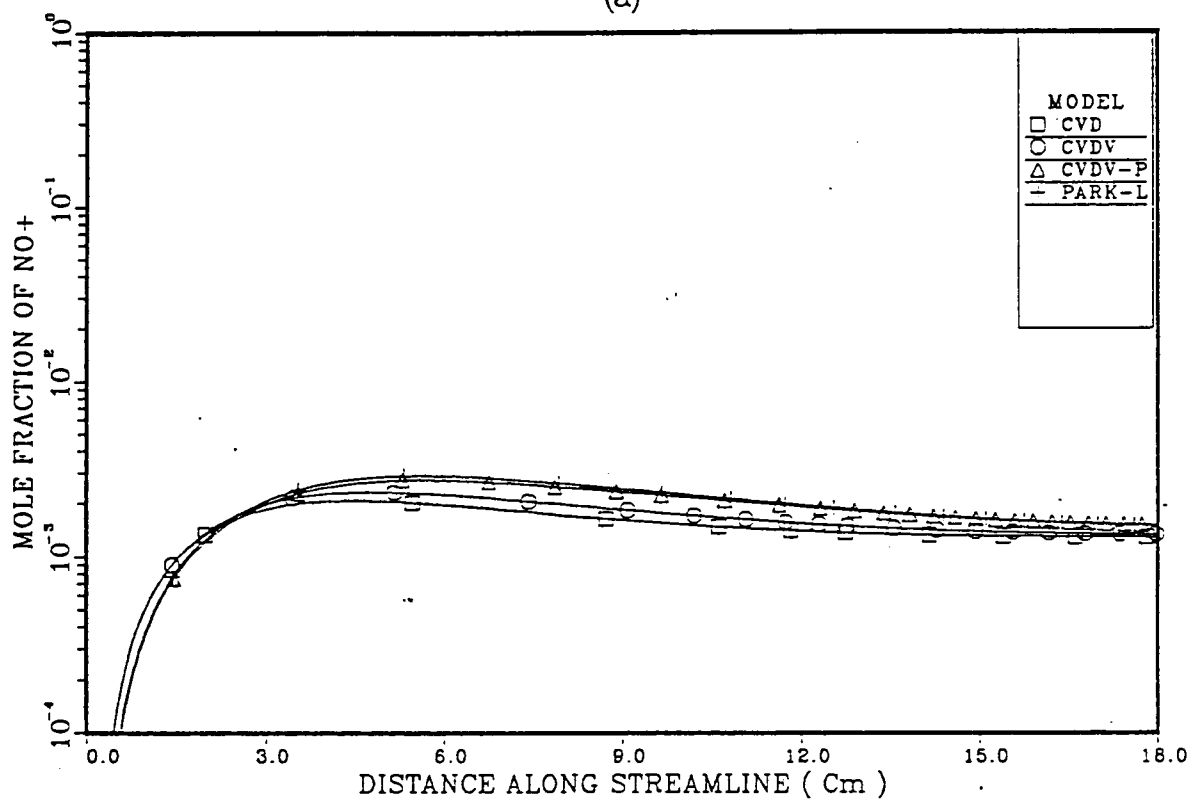


(b)

FIGURE 12. O AND NO PROFILES AT V=10 Km/s, RR1



(a)



(b)

FIGURE 13. NO^+ AND e^- PROFILES AT $V=10$ Km/s, RR1 10^{-4}

NO concentration profiles when they are compared against the results obtained from the various vibrational coupling models. In addition, the concentrations of N , NO^+ , and e^- predicted by the vibrational equilibrium model were very similar to those predicted by the other models.

As shown in Figure 6(b), the CVD model predicts that the N_2 vibrational temperature overshoots the translational temperature near the shock front. This phenomenon is a classic example of the consequences of not including the effects that dissociation and recombination have on the vibrational energy in a vibration-dissociation model. By examining Figure 10(b), it can be seen that the vibrational temperature of N_2 for CVD is higher than in the other models. The additional vibrational energy within the physical system tends to enhance the dissociation of the diatomic species and causes a more rapid increase in monatomic concentrations as compared to the other models. These trends can be seen in Figures 10(a) and 11(a).

On Figure 7(b), it can be seen that the CVDV model predicts a lower N_2 vibrational temperature profile than the translational temperature because it includes the changes in vibrational energy due to the effects of dissociation and recombination. These additional effects not only change the N_2 vibrational temperature profile, as seen in Figure 10(b), but also decrease the N_2 and O_2 dissociation rates [Figures 10(a), 11(b)]. This decrease in N_2 and O_2 dissociation delays the production of N [Figure 11(a)], but shows little effect

on the concentrations of the remaining species. From these trends, it can be concluded that the changes in concentration and temperature along a streamline are, for the (RR1) system, directly tied and sensitive to the N_2 vibrational energy and the vibration-dissociation coupling model.

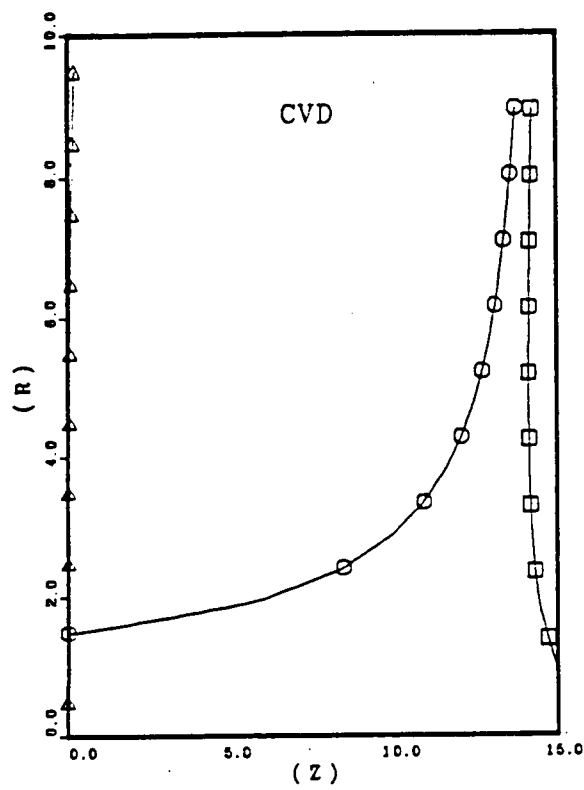
Since the CVDV-Preferential model only allows the upper vibrational energy levels to easily dissociate, and since these levels are less populated than the lower ones, it predicts a rate of dissociation even slower than the aforementioned models. This preferential coupling effect along with the corresponding decrease in N_2 dissociation, and the subsequent decrease in N production is very apparent on Figures 10(a), 11(a), and 11(b). The CVDV-Preferential model also displays a subsequent increase in the NO^+ and electron concentrations when compared to the CVD and CVDV models. This increase is depicted in Figures 13(a) and 13(b).

The results using the Park-Like model (Figure 9) show some interesting and significantly different trends when compared to the previously discussed models. As can be seen from Figures 9(b) and 10(b), the Park-Like model has an N_2 vibrational temperature profile which gradually rises to a peak value, that is approximately one third of the translational shock jump temperature, followed by a slow decrease to a constant or pseudo-equilibrium value. In addition, as shown on the concentration plots, the Park-Like model closely follows the concentration profiles associated with the CVDV-Preferential model for most of the species in

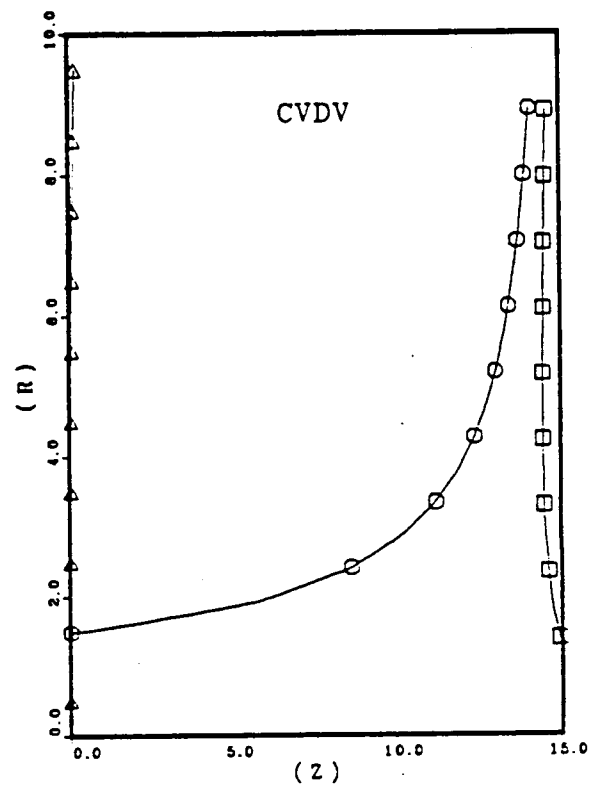
question. The only species component which shows non-preferential treatment near the shock front is O_2 ; but, as depicted in Figure 11(b), it equilibrates with the CVDV-Preferential model for the latter two thirds of the streamline. This correspondence is an unexpected result considering that the Park-Like model only modifies the N_2 vibration-dissociation interaction. Apparently, the modified Landau-Teller term, coupled with the additional CVDV terms, provides an additional mechanism similar to preferential control.

The final area of discussion for this trajectory point and reaction rate set is the geometric orientation of the shock, streamline, and body coordinates. These coordinates are depicted in Figure 14 for each vibration-dissociation coupling model. All of the vibrational coupling models basically show the same body shape and streamline orientation. As can be seen from Figure 14, the Maslen method breaks down near the stagnation region in its prediction of the body shape. This failure is due to the fact that the method assumes a small change in v-velocity with respect to x, which of course breaks down in the stagnation region.

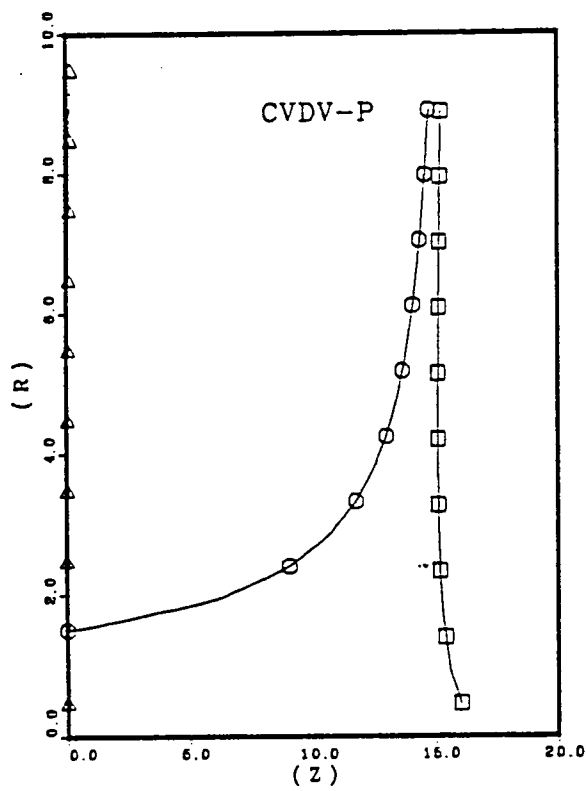
Nevertheless, the results for the streamline depicted in Figure 14 are valid. Another noticeable trend in the figure is the change in shock standoff distance between models, which shows that as dissociation processes are reduced, the shock standoff distance increases. This trend makes sense because as the dissociation processes are slowed down, the



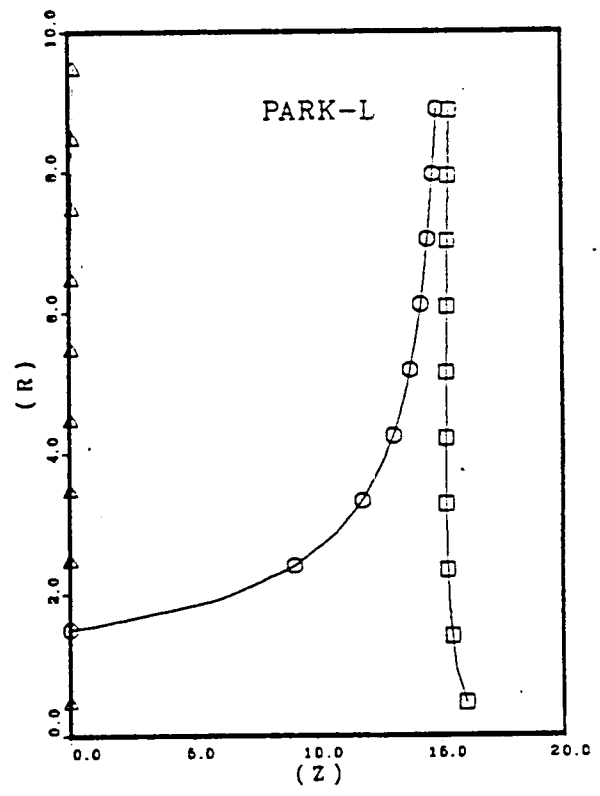
(a)



(b)



(c)



(d)

FIGURE 14. COORDINATES, $V=10$ Km/s, RR1

translational temperature decreases and the density increases at a slower rate. Therefore, the standoff distance must increase in order to maintain mass flux conservation.

An analysis of concentration and temperature profiles for the various vibration coupling models has also been conducted at the max-Q and exit trajectory points. Interestingly, all aforementioned trends for each species, temperature, and vibrational coupling model are the same. It appears that the trends in temperature and species concentration are relatively independent of freestream velocity for this reaction rate set. In addition, the streamline and body coordinates for both trajectory points maintained their orientation and body shape.

Reaction Rate Set 2.

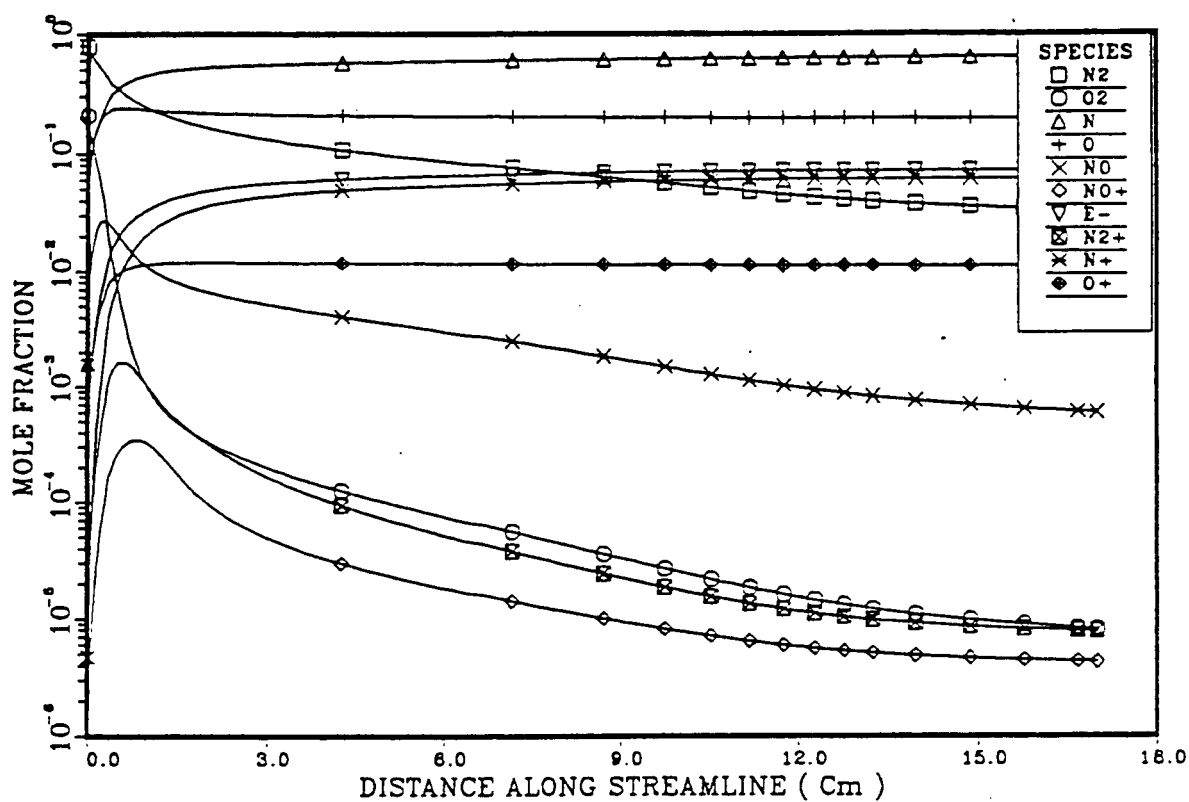
Reaction rate set 2 (RR2) contains the primary dissociation reactions of the first reaction rate set plus five additional chemical reactions^{18,19}. These additional reactions are included in order to further describe the ionization processes that take place within the shock layer. The physical system for (RR2) has been modeled by the ten species N_2 , O_2 , N , O , NO , NO^+ , e^- , N_2^+ , N^+ , and O^+ . In this set, the nitrogen ionization reaction rate coefficients¹⁸ are based on experimental data and assume that the ionization proceeds from the initial N ground state to the intermediate $3s^4P$ excited state followed by rapid ionization. It is also

assumed that the five additional reaction rate coefficients are reasonably accurate when extrapolated to temperatures beyond their experimental range. In addition, for the electron impact reaction ($N + e^- \rightleftharpoons N^+ + 2e^-$), the rate coefficient is assumed to be governed by electron temperature instead of heavy particle or translational temperature. In this program, this reaction is actually controlled by the N_2 vibrational temperature since the electron temperature has been modeled by T_{vn2} in the present study.

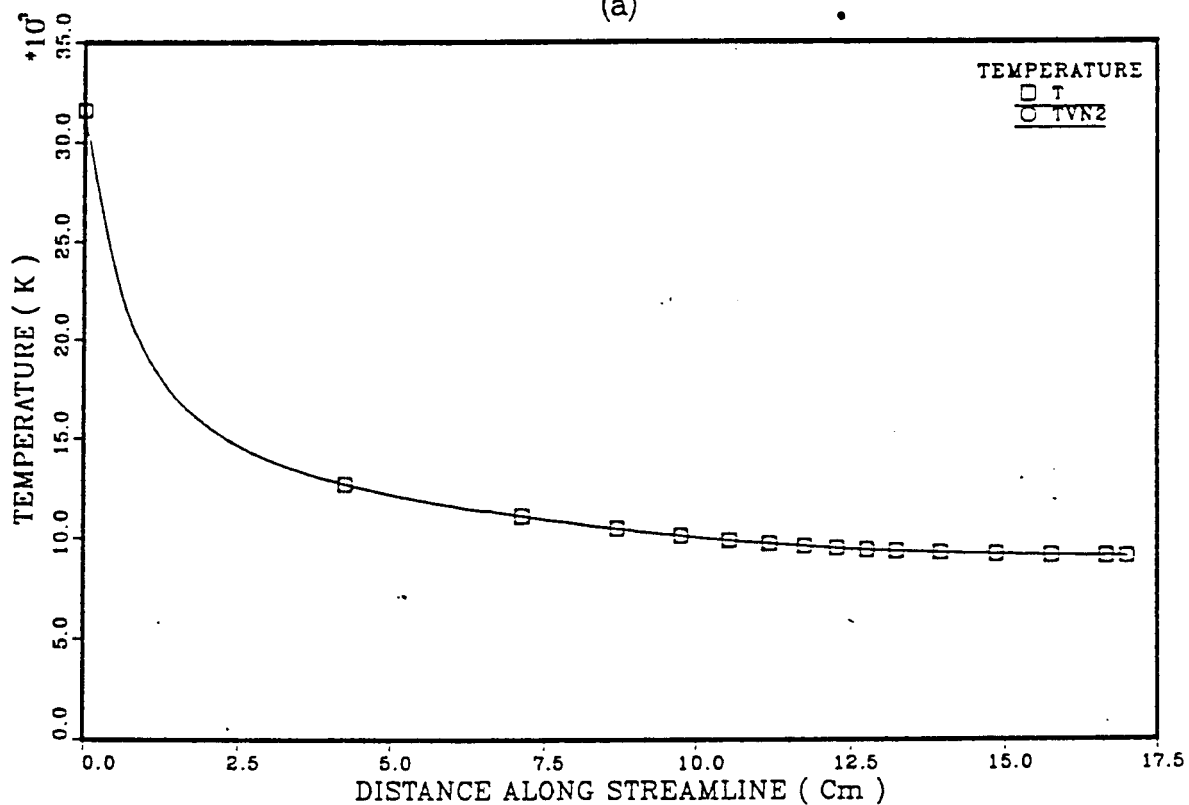
The concentration and temperature profiles for each vibration coupling model are plotted and displayed at the entry trajectory point in Figures 15-19. In addition, a comparative study has again been made concerning the vibration coupling model effects on vibrational temperature and species concentrations as depicted in Figures 20-25.

As can be seen in Figure 15, the vibrational equilibrium case exhibits a temperature profile similar to that seen with the first reaction rate set. Since the first reaction rate set contains only one ionization reaction, the NO^+ and e^- concentrations are always in equilibrium with each other due to the conservation of mass. The additional ionization reactions in (RR2), however, produce an increase in the amount of electrons, which leads to considerable changes in the ion species concentrations.

Figure 16 shows that, for this reaction set, the CVD model exhibits an N_2 vibrational temperature overshoot of the translational temperature, which is similar to the improper

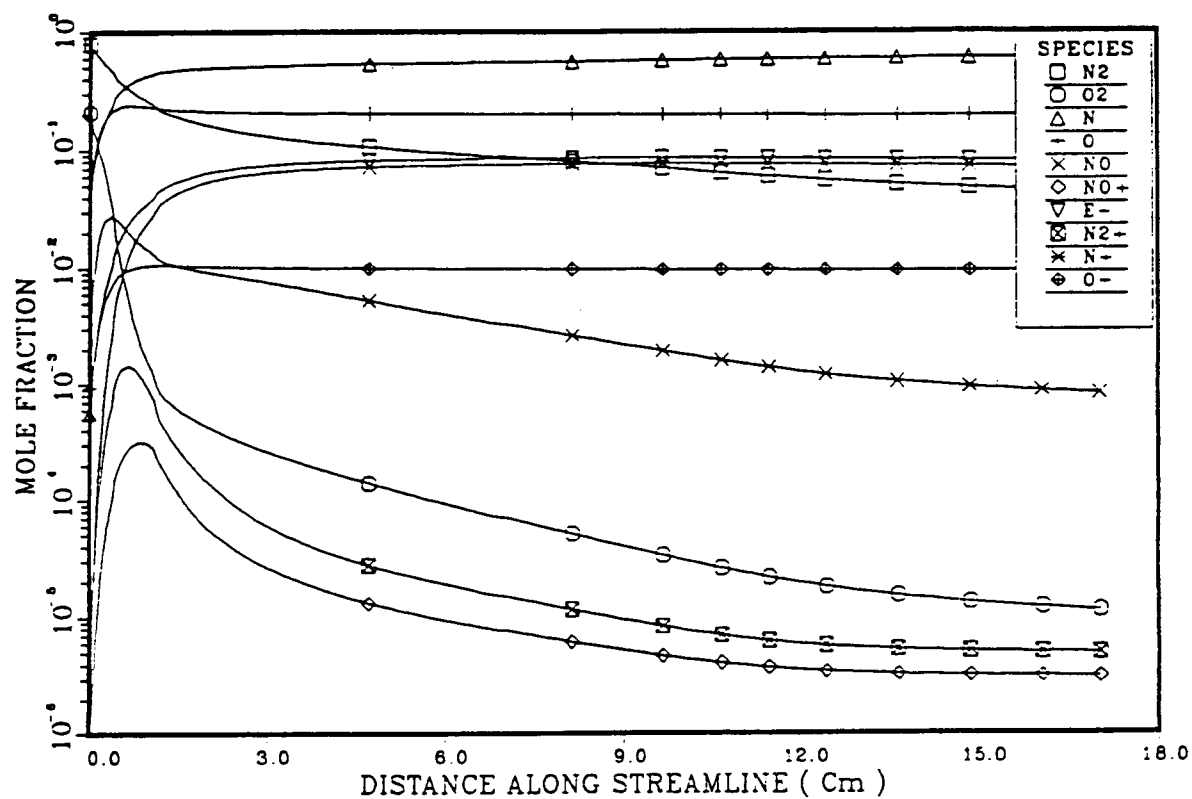


(a)

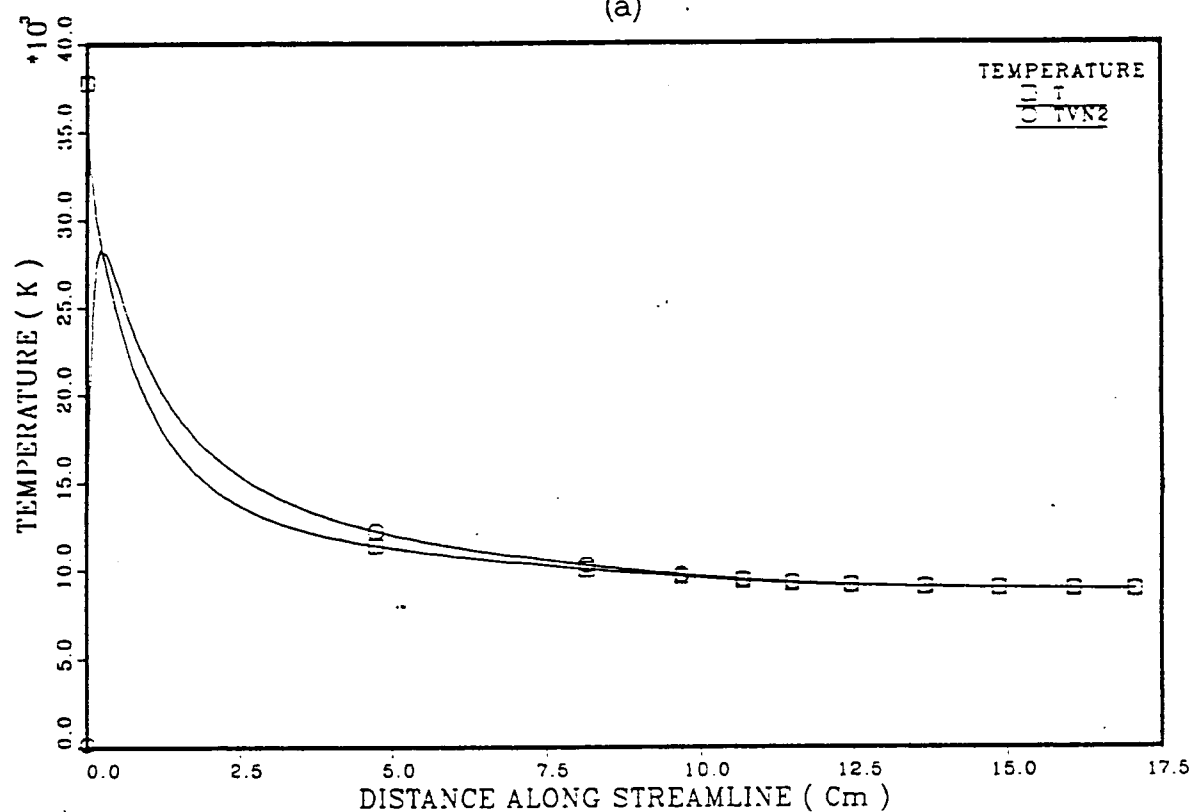


(b)

FIGURE 15. VEQ MODEL AT $V=10$ Km/s, RR2

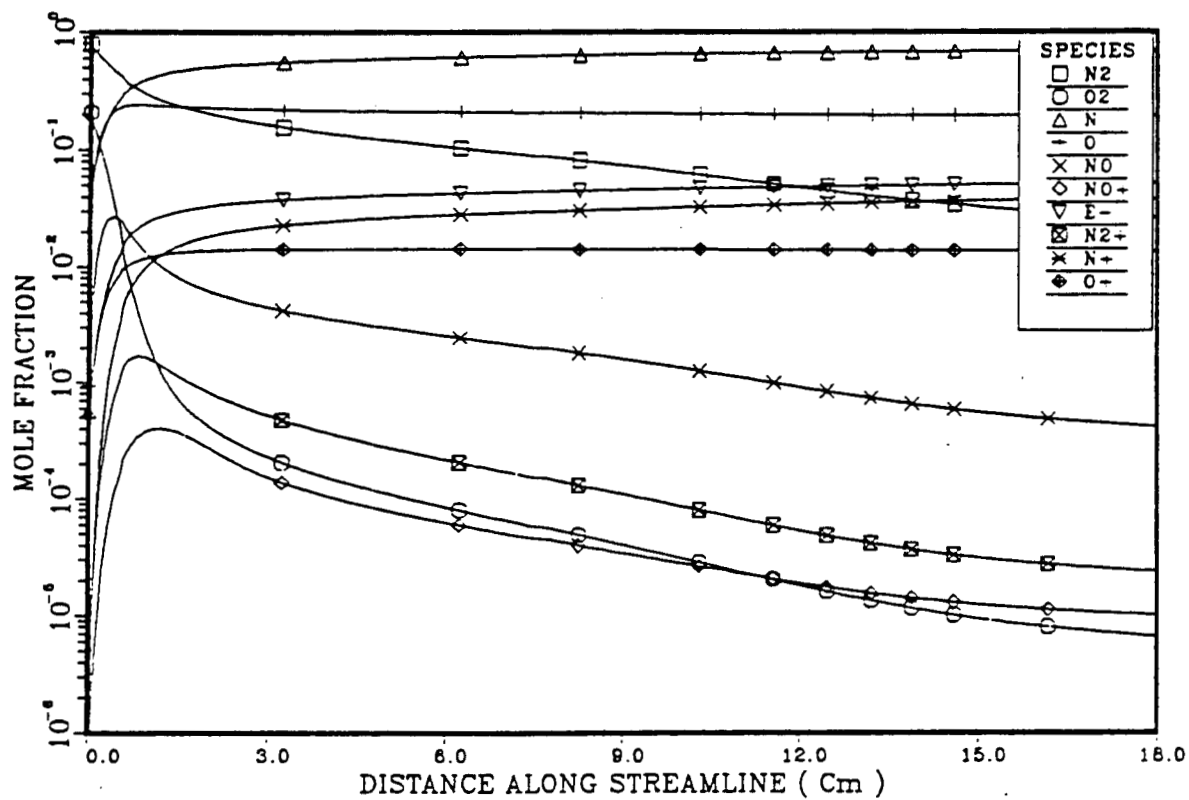


(a)

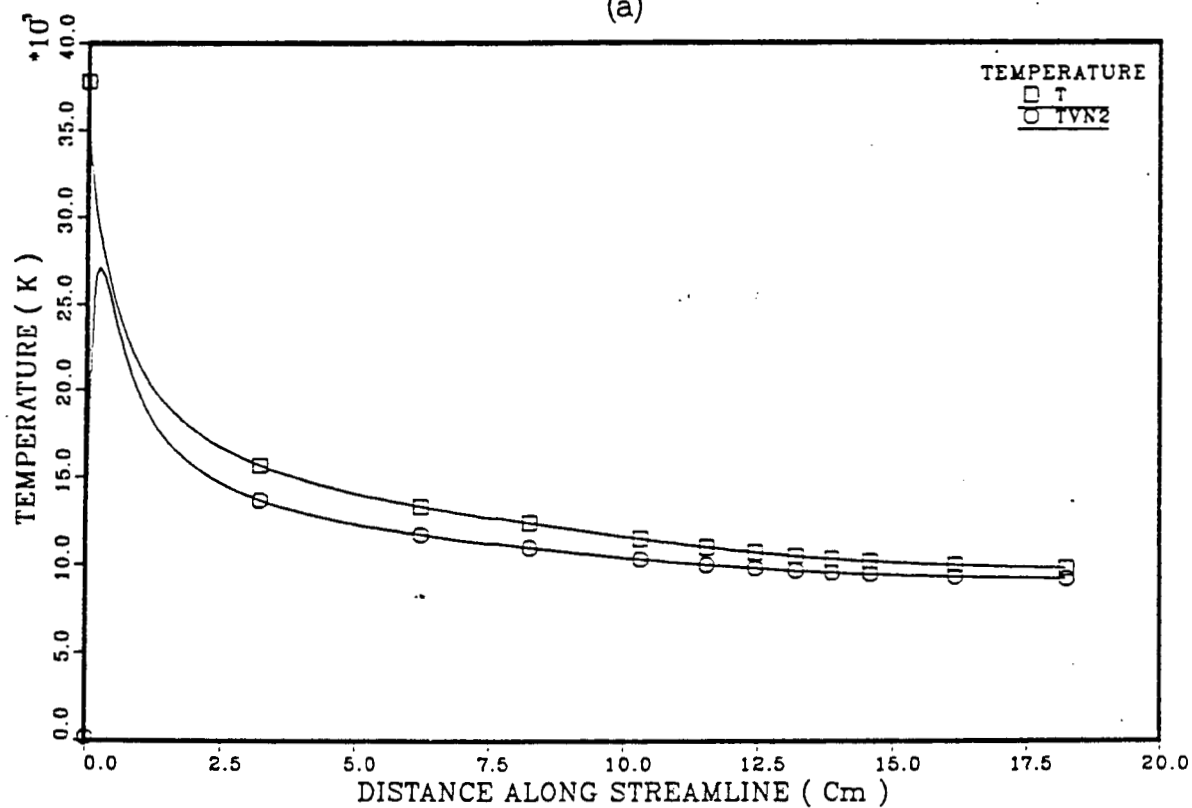


(b)

FIGURE 16. CVD MODEL AT V=10 Km/s, RR2

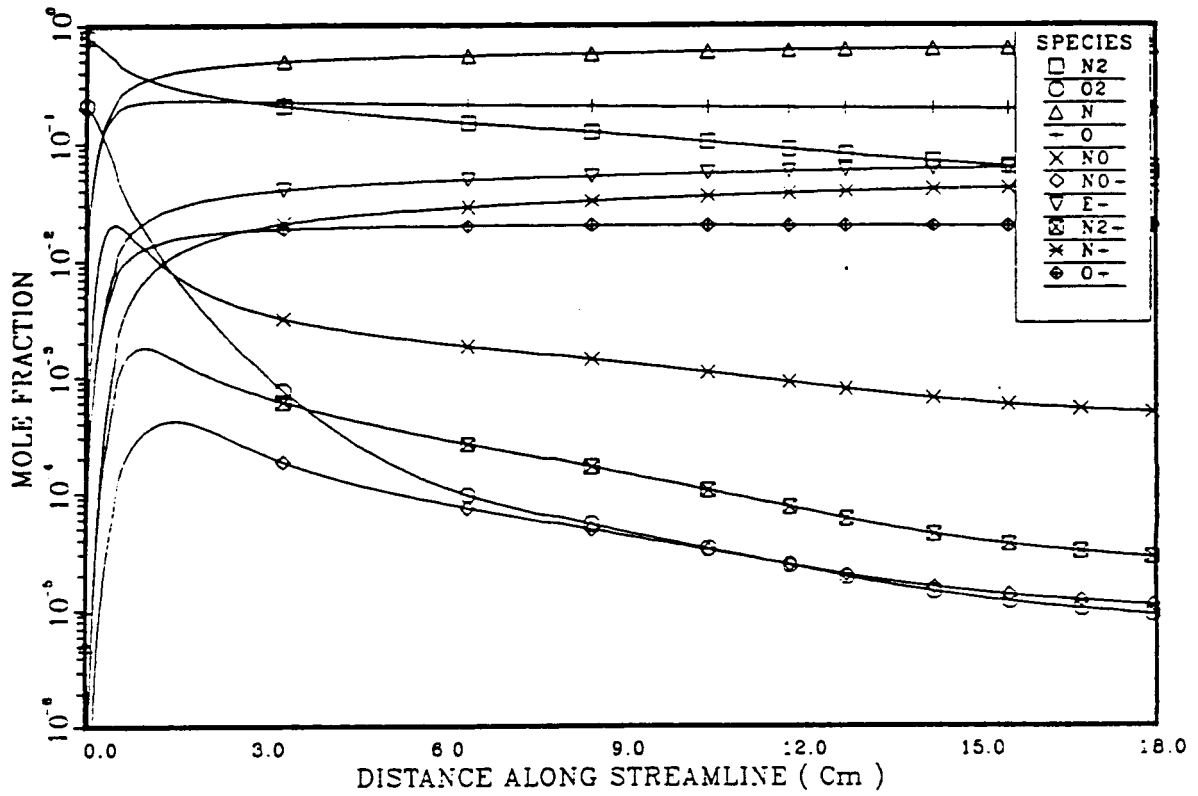


(a)

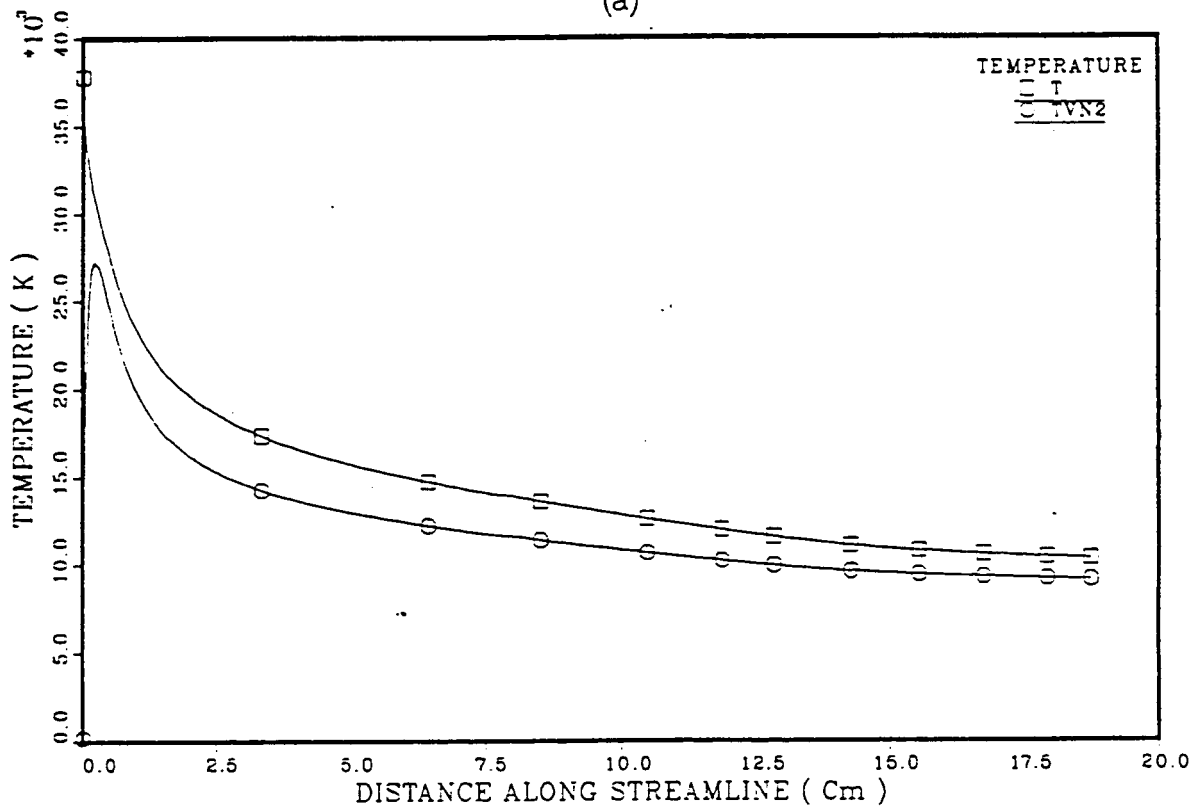


(b)

FIGURE 17. CVDV MODEL AT $V=10$ Km/s, RR2

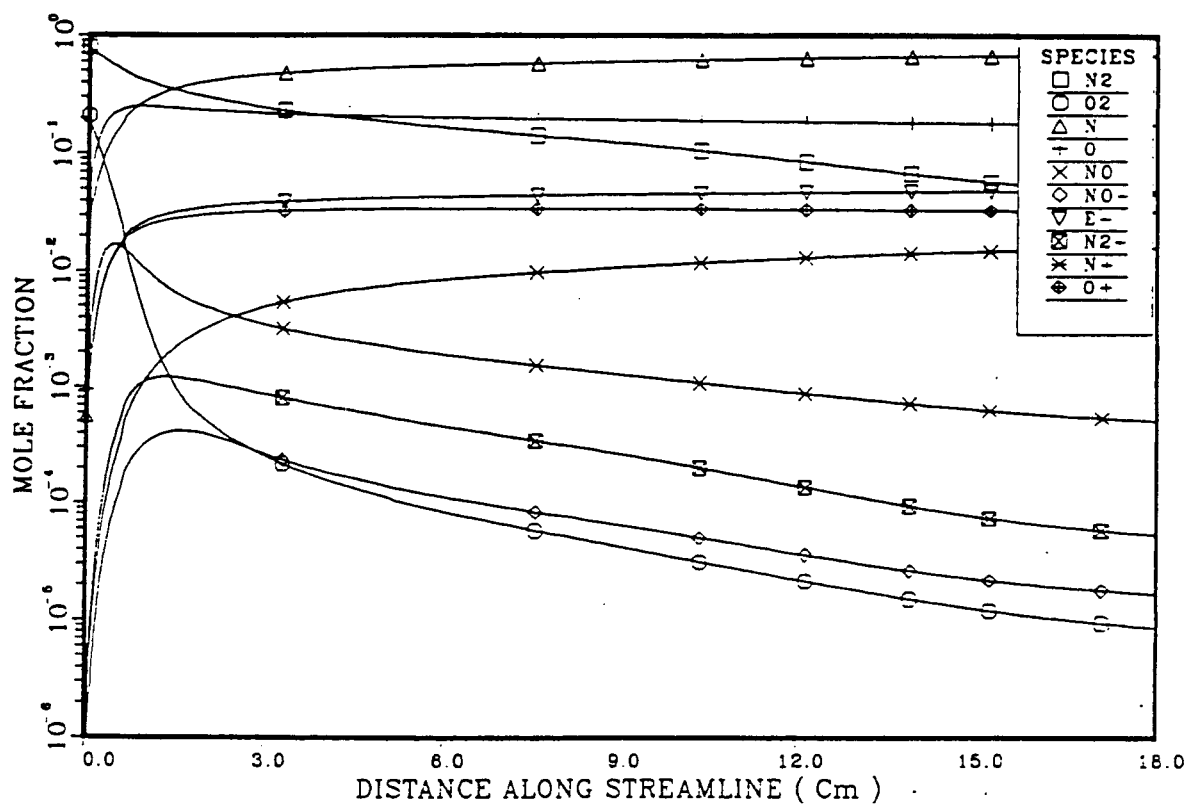


(a)

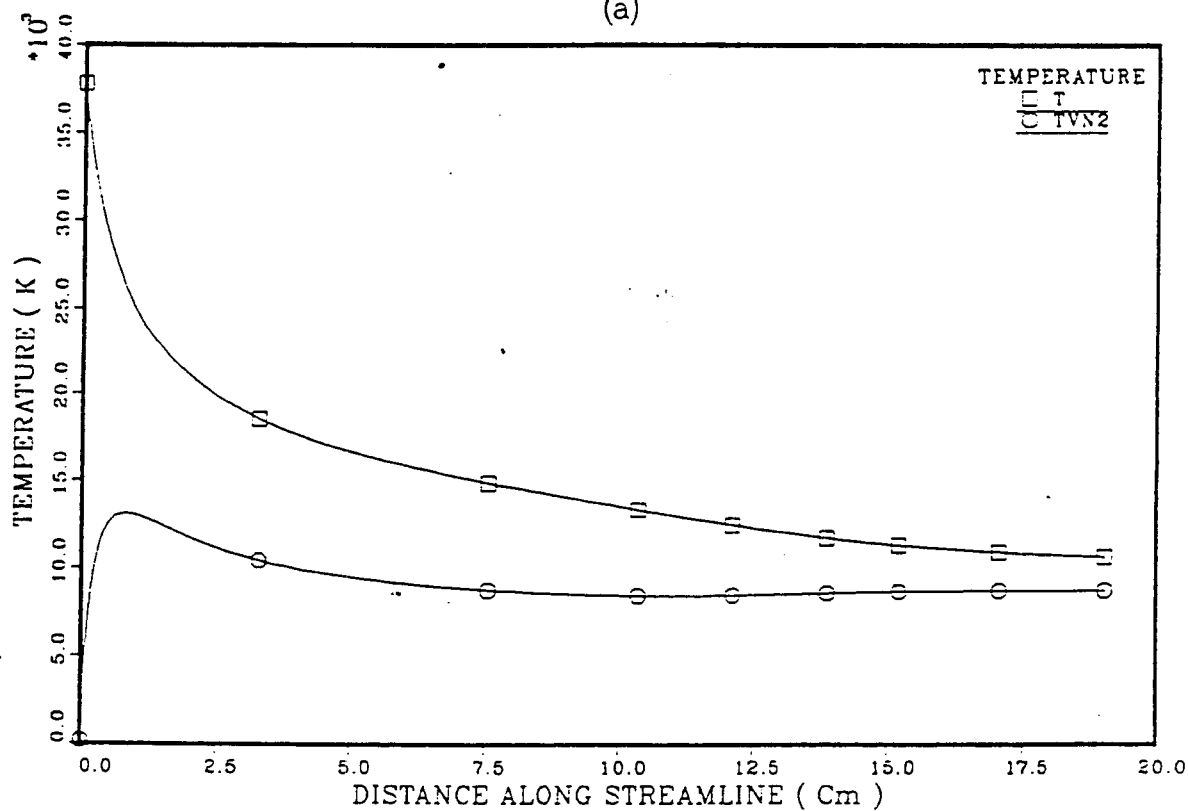


(b)

FIGURE 18. CVDV-P MODEL AT $V=10$ Km/s, RR2

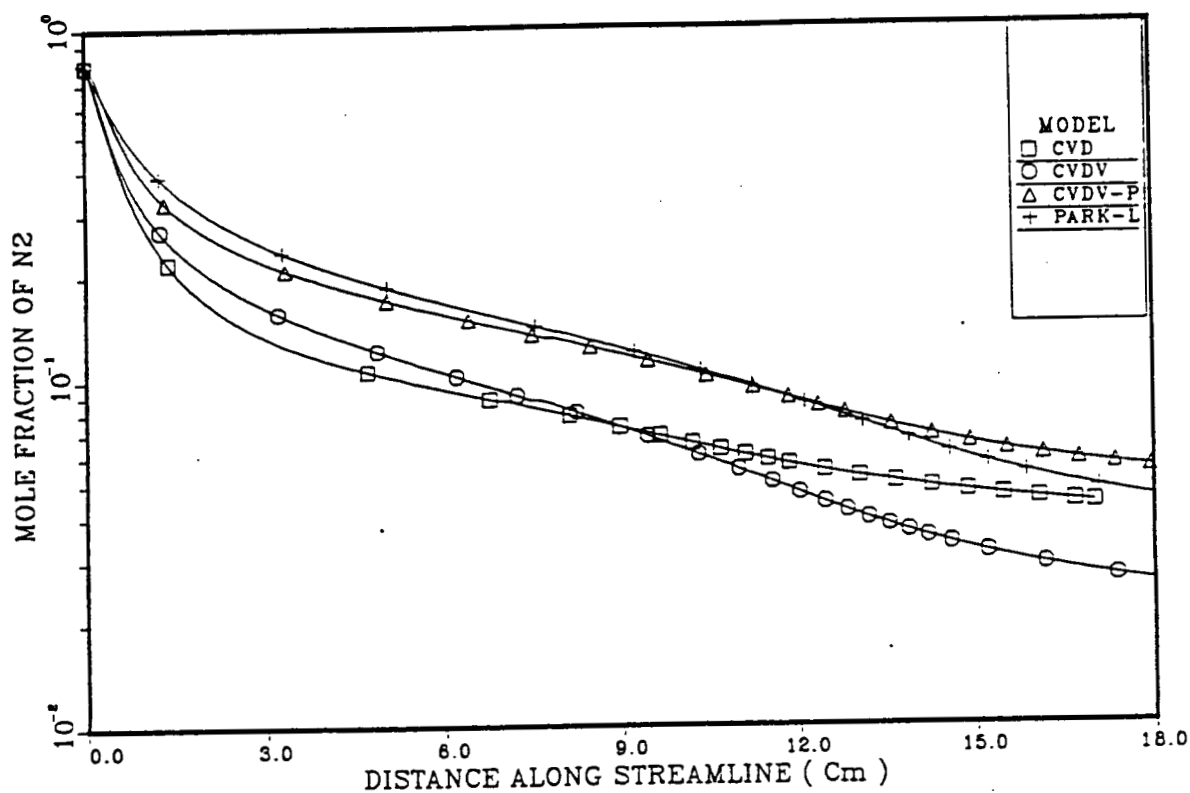


(a)

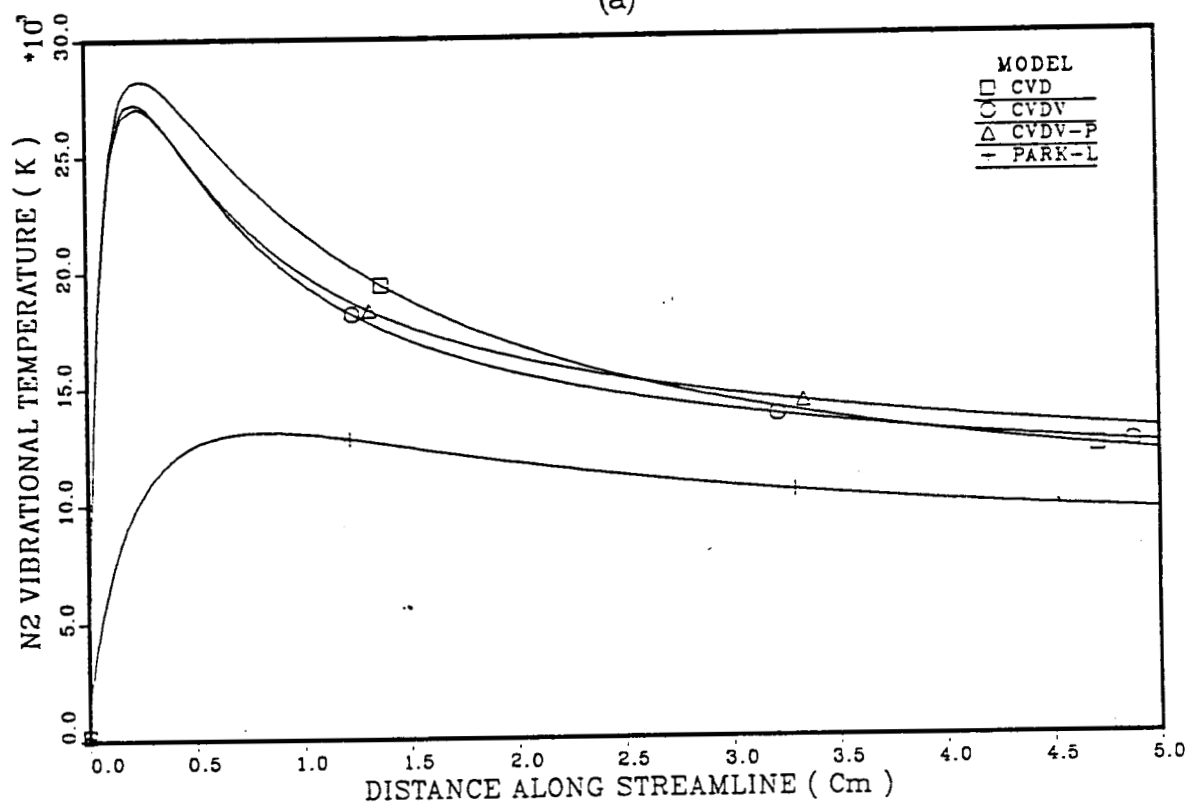


(b)

FIGURE 19. PARK-L MODEL AT $V=10$ Km/s, RR2

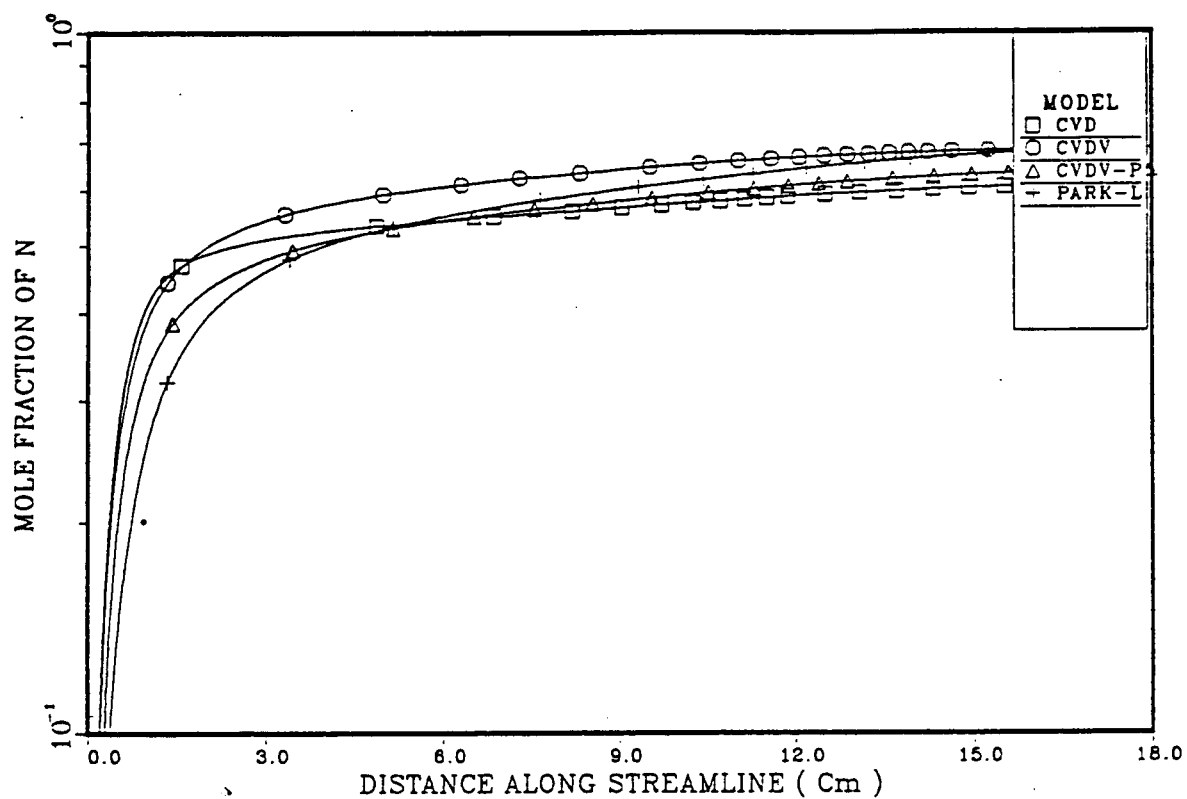


(a)

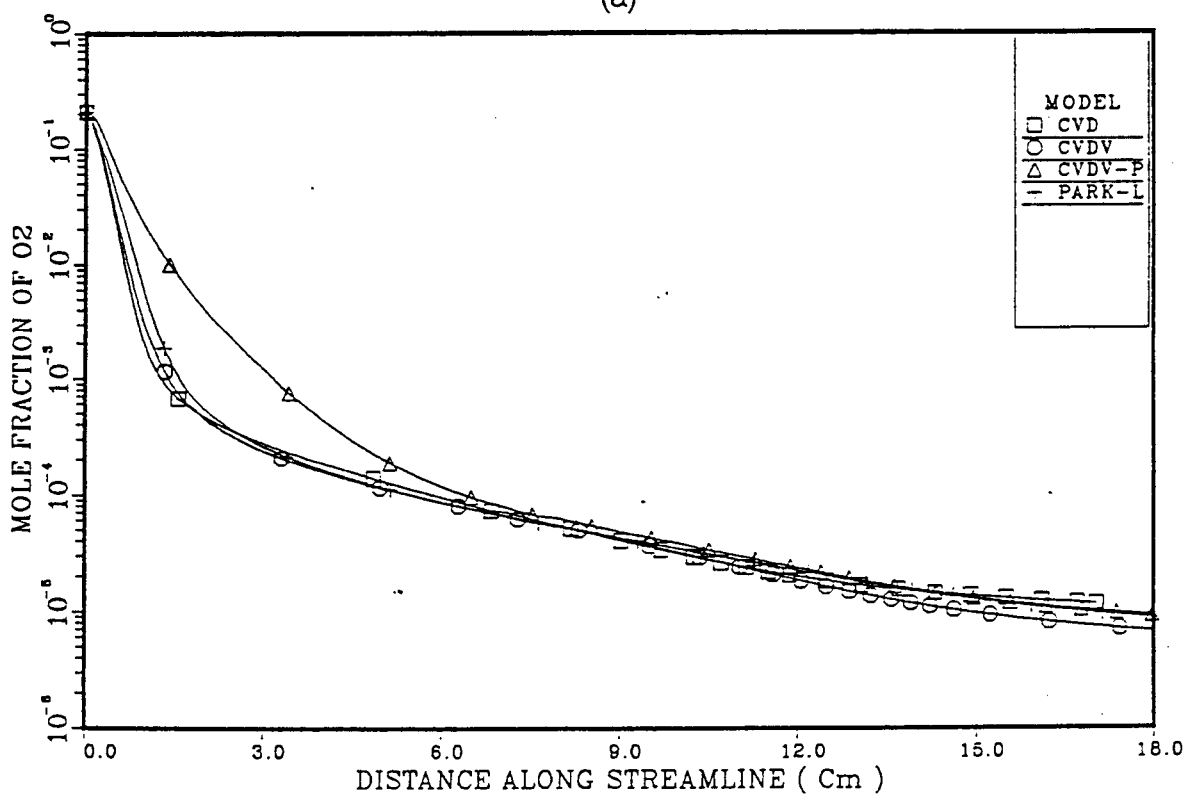


(b)

FIGURE 20. TVN2 AND N2 PROFILES AT V=10 Km/s, RR2

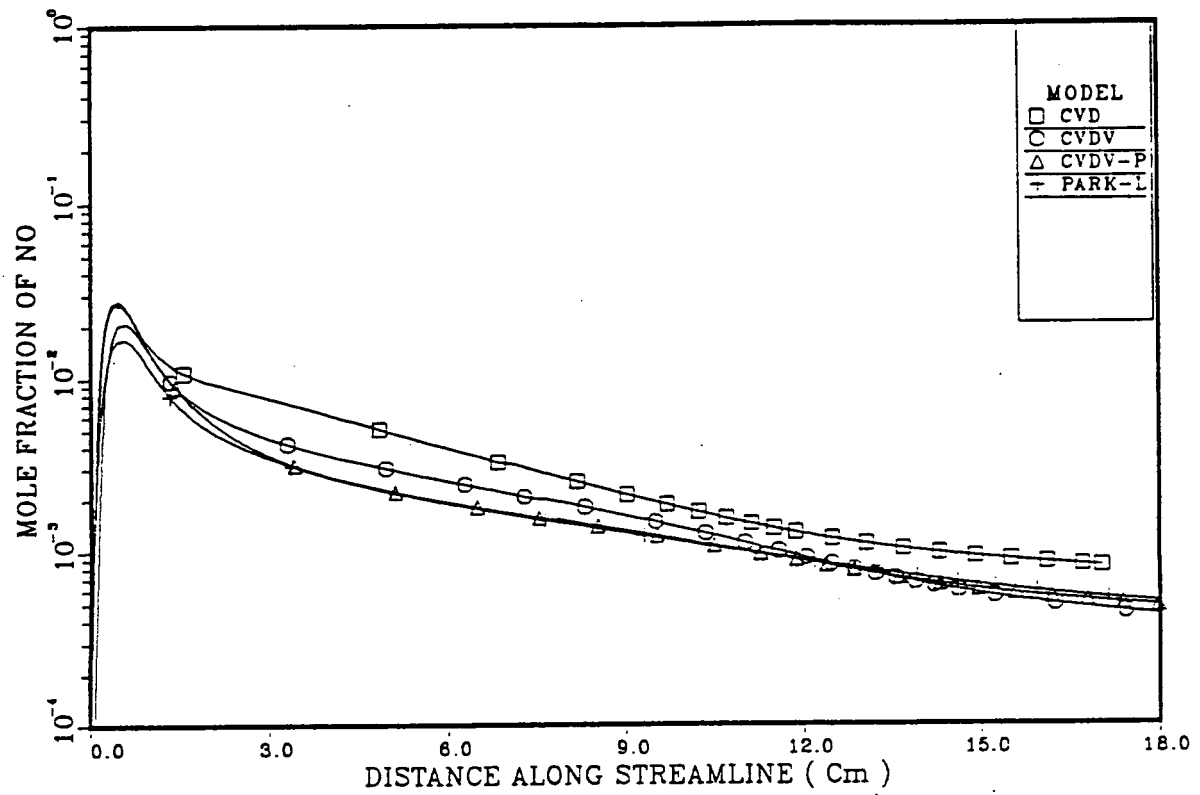


(a)

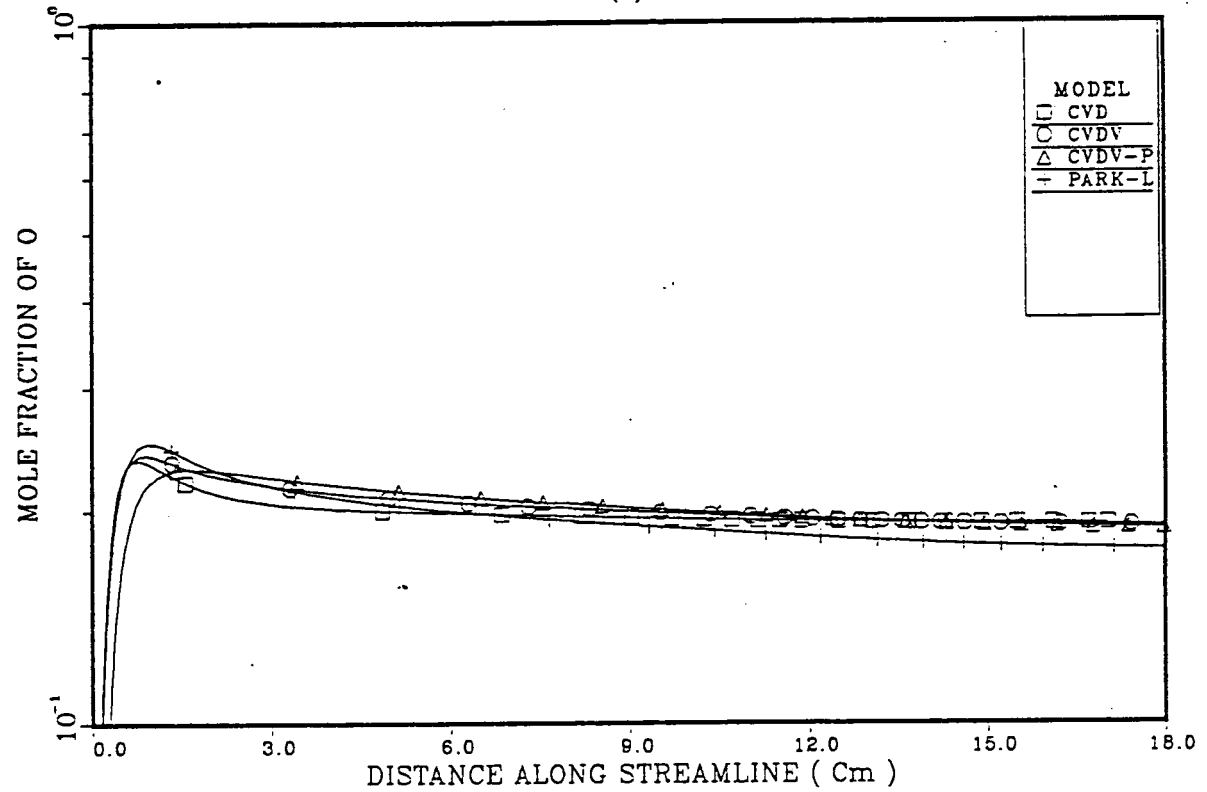


(b)

FIGURE 21. O2 AND N PROFILES AT V=10 Km/s, RR2

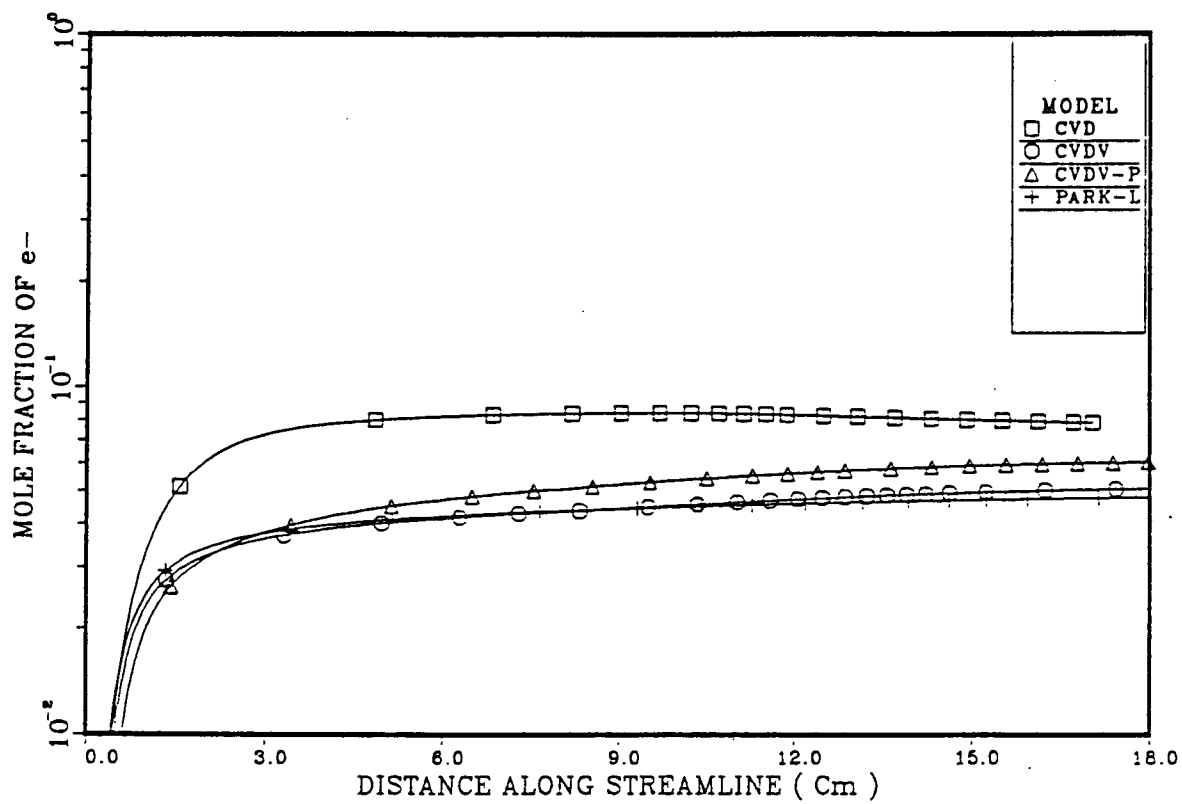


(a)

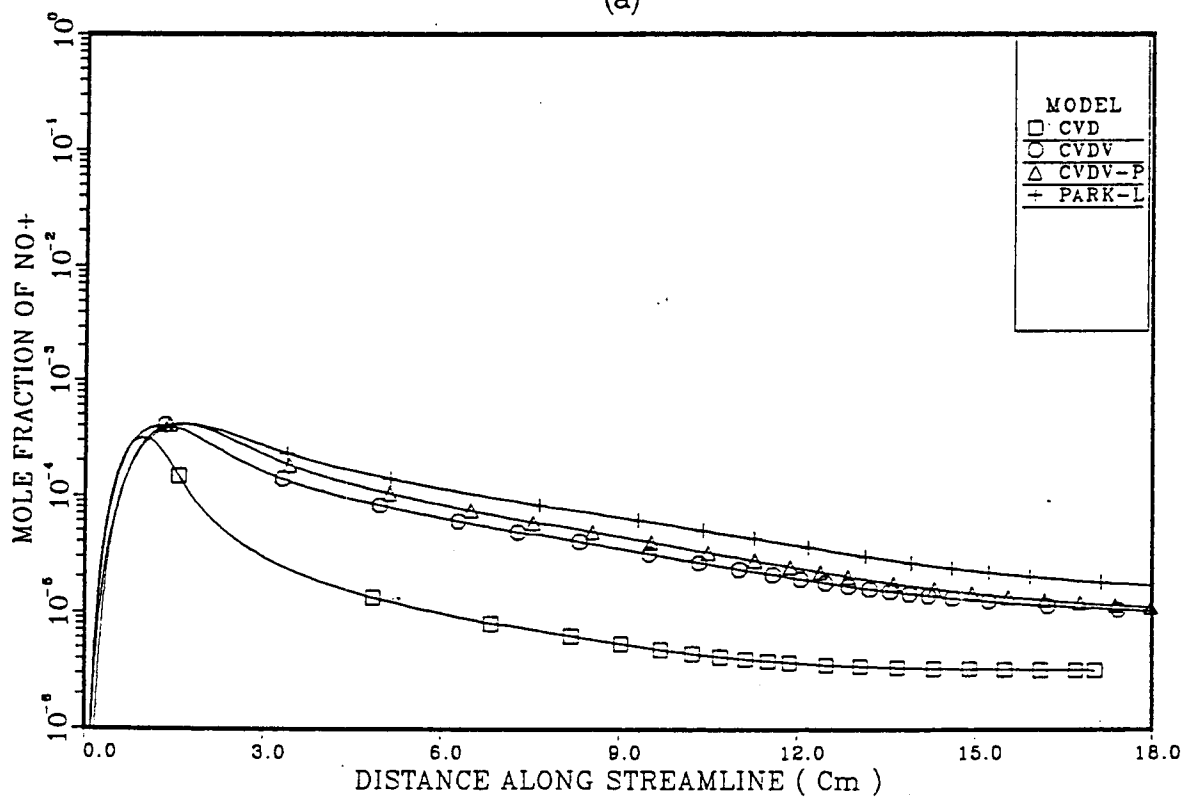


(b)

FIGURE 22. O AND NO PROFILES AT V=10 Km/s, RR2

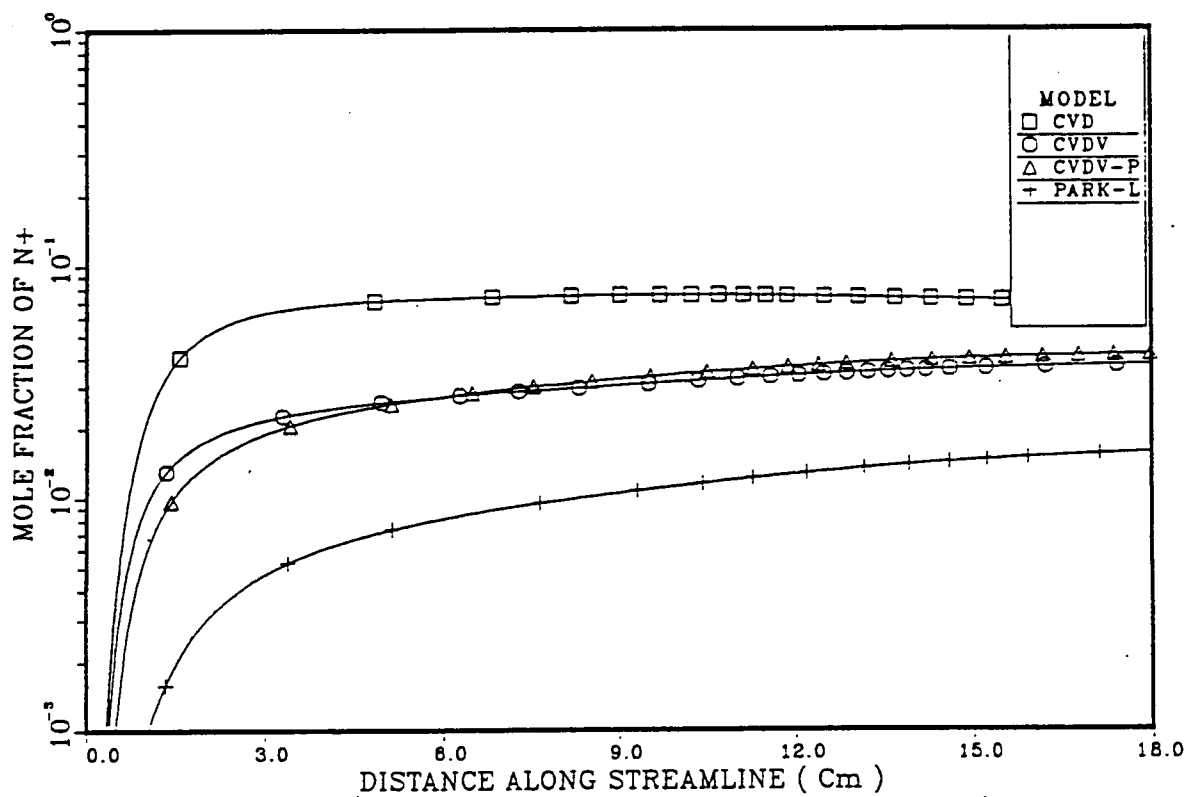


(a)

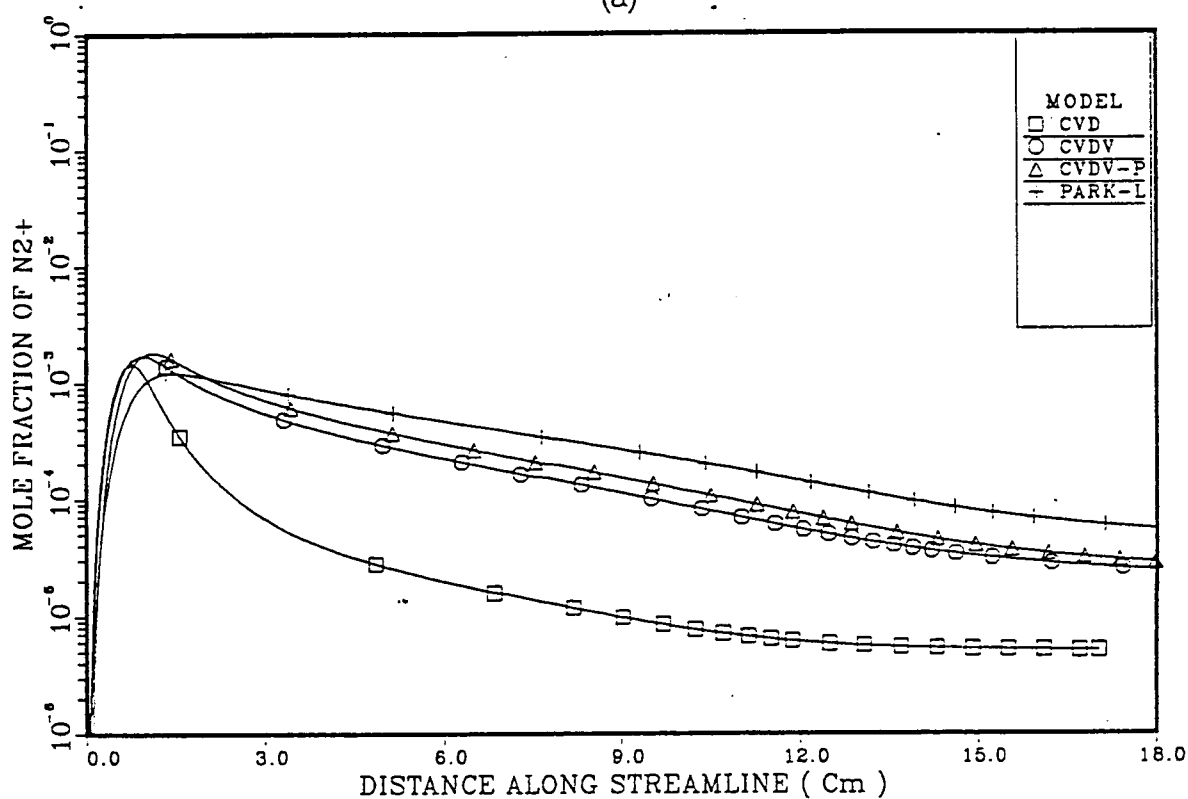


(b)

FIGURE 23. NO^+ AND e^- PROFILES AT $V=10$ Km/s, RR2



(a)



(b)

FIGURE 24. N_2^+ AND N^+ PROFILES AT $V=10$ Km/s, RR2

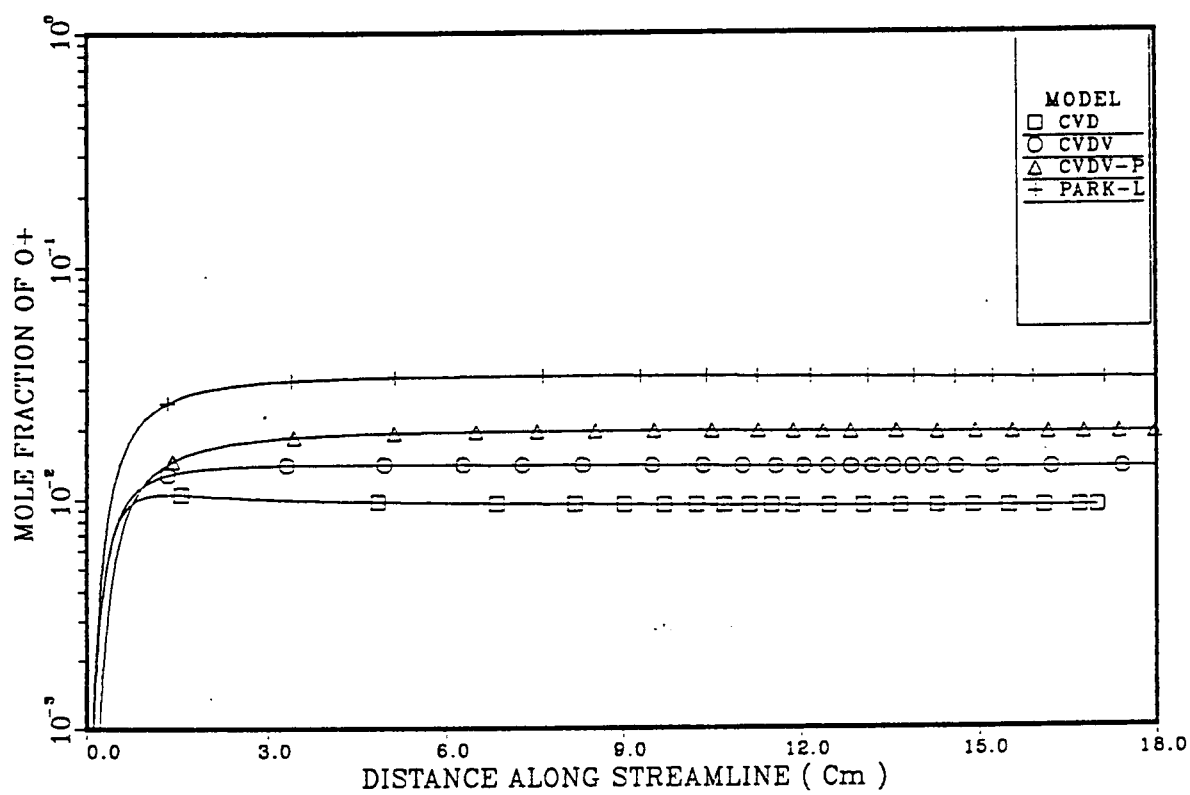


FIGURE 25. O+ PROFILE AT V=10 Km/s, RR2

behavior observed with the previous reaction rate set. Also, the predicted initial electron production for the CVD model is slightly greater than that of the vibrational equilibrium case. But as seen in Figure 23(a), the electron concentration is significantly greater than that predicted by the other vibrational coupling models. Furthermore, it can be seen from the model comparison plots that the CVD model predicts a greater concentration of N^+ ions and a subsequent decrease in other ion formations as compared to the aforementioned models.

The CVDV model results, which are plotted on Figure 17, display a rapid increase in N_2 vibrational temperature near the shock front followed by a gradual decrease that parallels the translational temperature. This vibrational temperature profile exhibits the same trend as predicted by the previous reaction rate model. Interestingly, the CVDV model displays a significant decrease in electron production as compared to the CVD model (Figure 23). This decrease in electron production is accompanied by a corresponding decrease in the N^+ concentration and an increase in the NO^+ , N_2^+ , and O^+ concentrations. Considering that the atomic nitrogen concentration prediction from the CVDV model is higher than the CVD prediction, as shown in Figure 21(a), these charged species results are somewhat surprising. These trends indicate a significant interaction between vibration-dissociation coupling processes and the ionization reactions.

For the CVDV-Preferential model, the temperature profile

of Figure 17(b) shows a similar pattern to that predicted by the first reaction rate model. The electron production is slightly greater than that of the CVDV model for the same reaction rate model; and therefore, the preferential results also depicts similar changes in ion concentration. As expected, with the exception of the O_2 species, the rates of dissociation are slightly slower than that predicted by the CVDV model. The O_2 species displays significant reductions in initial dissociation due to preferential treatment and control.

The temperature and concentration results from the Park-Like model are shown in Figure 19 are similar to those obtained by the first reaction rate model. Again, as shown in Figures 20-22, the dissociation process and concentration profiles for the non-ionized species closely follow the concentrations predicted by the CVDV-Preferential model. However, the ionization profiles on Figures 23-25 are different than those obtained with the CVD, CVDV, and CVDV-Preferential models. Since the electron impact reaction rate ($N + e^- \rightleftharpoons N^+ + 2e^-$) is controlled by T_{vn2} , which is shown as being lower for the Park-Like model [Figure 20(b)], the production of N^+ is significantly reduced. This reduction in the amount of N^+ is accompanied by subsequent increases in N_2^+ , NO^+ , and O^+ when compared to the other vibrational coupling models. Surprisingly, the increase in the electron production rate is significant enough to allow a greater mole fraction concentration of O^+ over that of N^+ . The Park-Like

model is the only model to display this characteristic for this trajectory point.

For this trajectory point and reaction rate set, the shock, streamline, and body coordinates, which are shown on Figure 26, display the expected changes in shock standoff distance due to vibration-dissociation coupling model differences. In addition differences become apparent, when comparing (RR1) to (RR2), with the same vibrational coupling model. The standoff distances for (RR2) are actually less for each model even though the trends between the models remain the same. Apparently, the additional species and ionization reactions of (RR2) increase the amount of ionization, and decrease the temperature, which in turn leads to an increase in density and a shorter standoff distance.

Results have also been obtained with this reaction rate model at the max-Q and exit trajectory points. While the aforementioned temperature, dissociation, and ionization trends remained the same for both points, the results did, of course, show a reduction in temperature and production rates associated with the reductions in freestream velocity. The only distinguishable trend that was different from the entry trajectory point was a greater concentration of O^+ over that of N^+ for all vibrational coupling models. This trend was coupled to an order of magnitude reduction in electron concentration and a decrease in the vibrational temperature of nitrogen. It would appear that sufficient magnitudes of both electron concentration and N_2 vibrational temperature

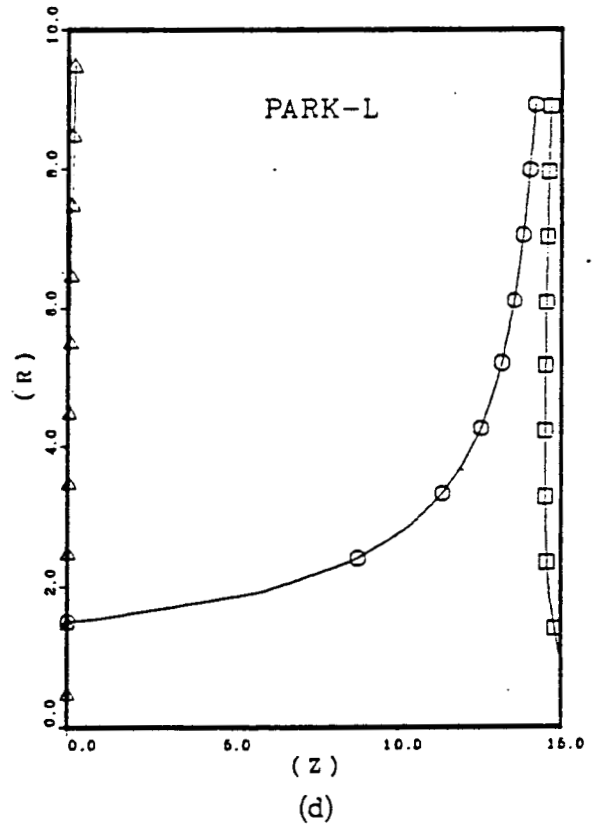
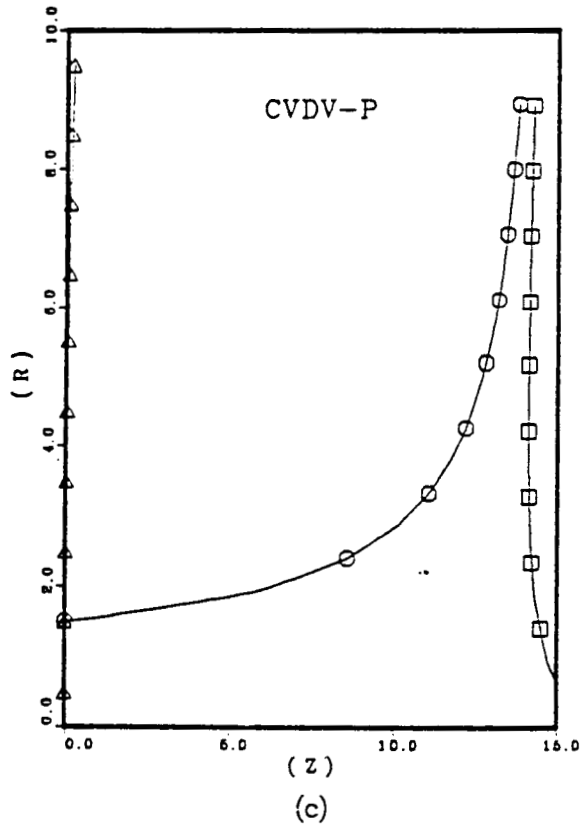
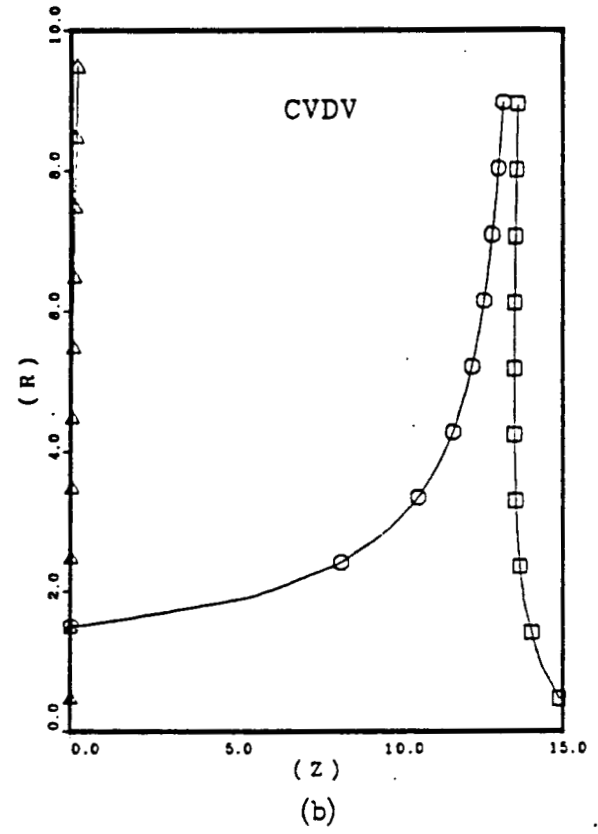
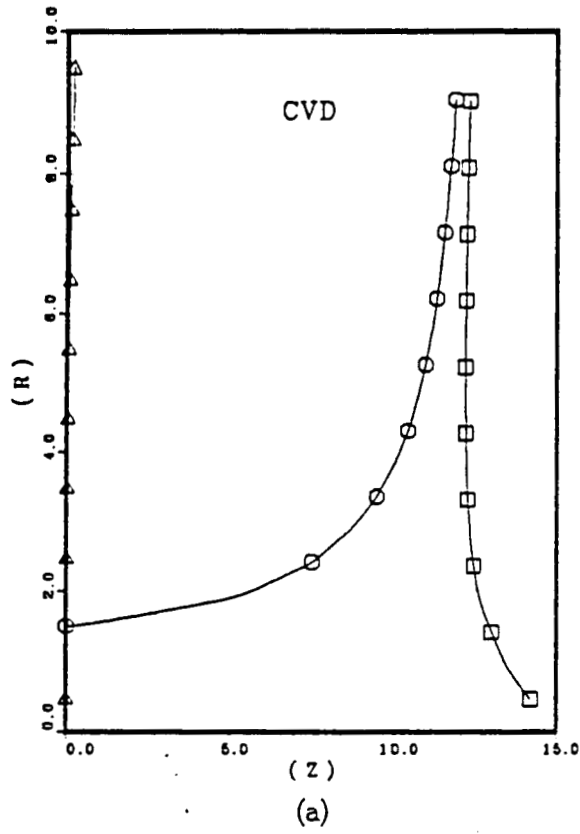


FIGURE 26. COORDINATES, $V=10$ Km/s, RR2

125

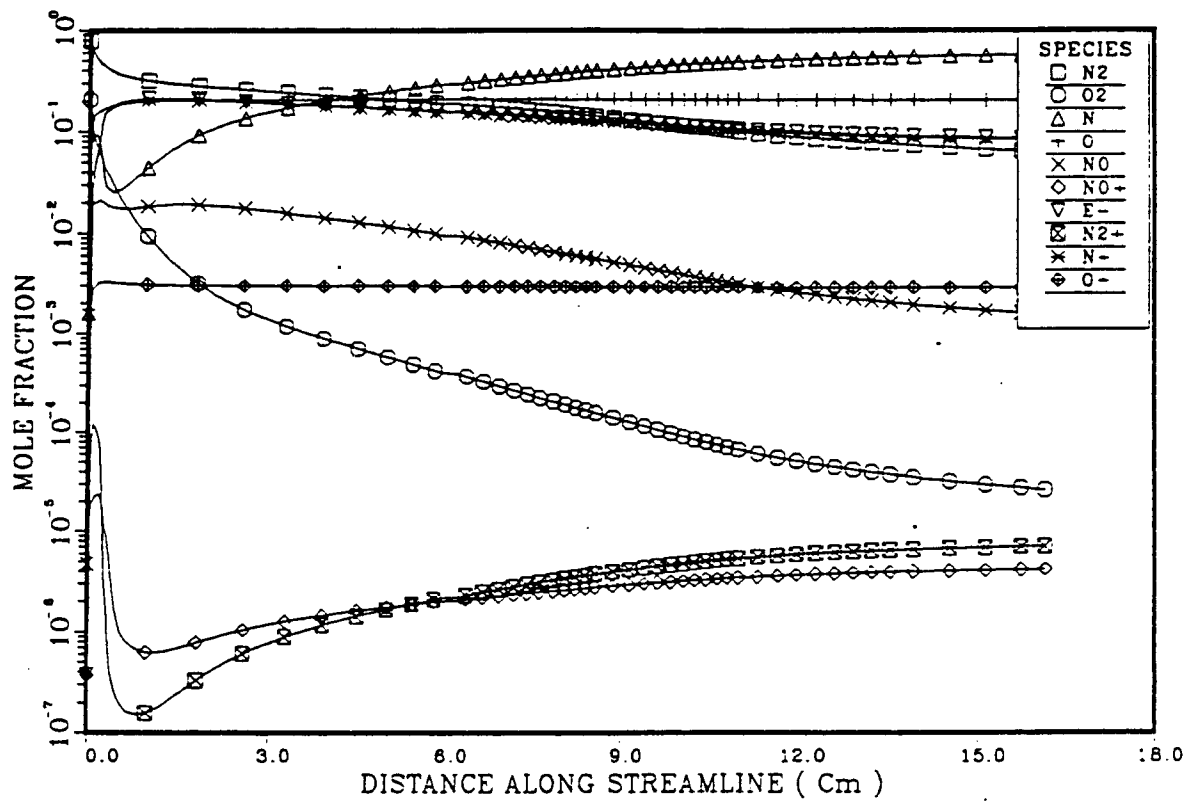
ORIGINAL PAGE IS
OF POOR QUALITY

are necessary conditions for a dominant N^+ population to exist.

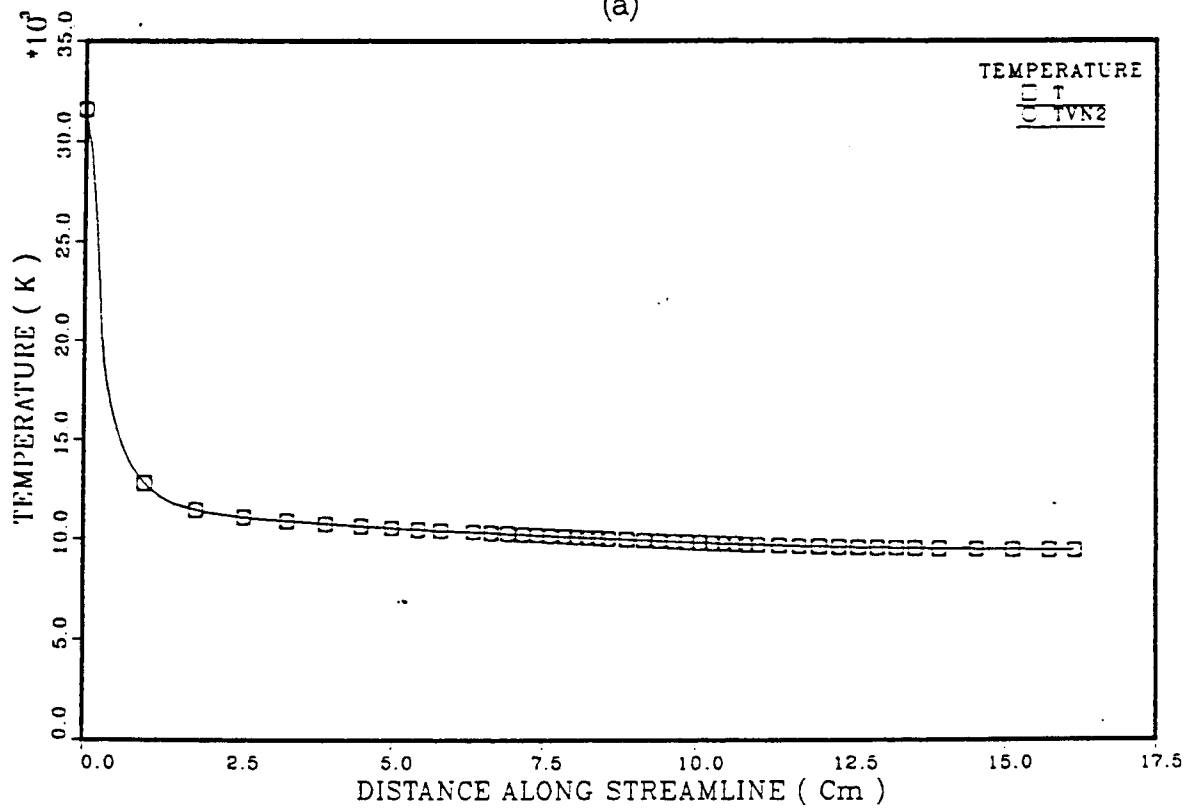
Reaction Rate Set 3.

Reaction rate set 3 (RR3) also includes ten species with the six primary reactions of the first reaction rate set plus the five additional ionization reactions. The five additional reactions are the same as those described in the second reaction rate set, except that the reaction rate coefficients for the electron impact reaction²⁰ have been changed. While the electron impact reaction rate is still being controlled by the electron temperature (or vibrational temperature of N_2), it is based on a one step process instead of a assumed two step intermediate sequence as previously discussed with the second reaction rate set. In addition, the updated electron impact reaction rate is considerably faster than the previous impact rate. Again, this reaction rate set is assumed to be reasonably accurate when extrapolated to higher temperatures. As with the previous reaction rate sets, the results for the entry trajectory point are plotted and displayed in Figures 27-37.

The vibrational equilibrium case, as shown in Figure 27, depicts a faster rate of equilibration than the CVD, CVDV, CVDV-Preferential and Park-Like models. Also, it can be seen that the usage of the faster electron impact ionization rate produces a very rapid increase in electron concentration

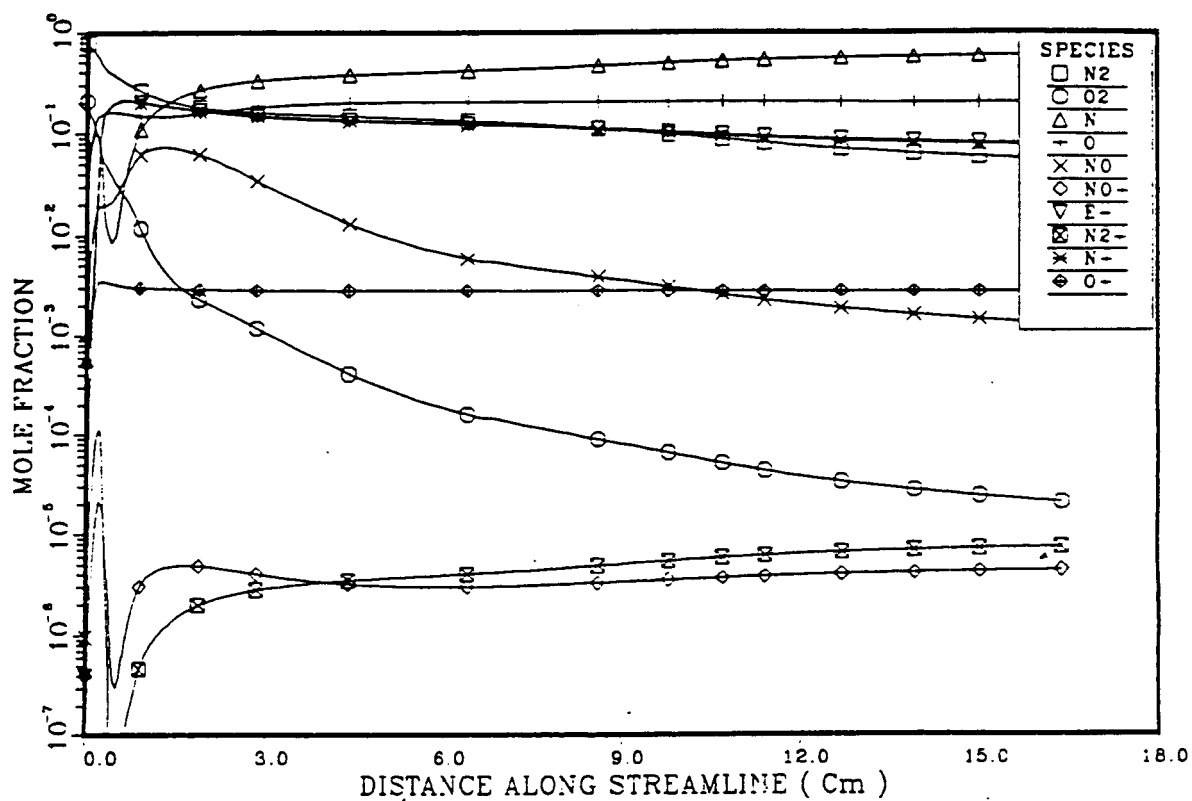


(a)

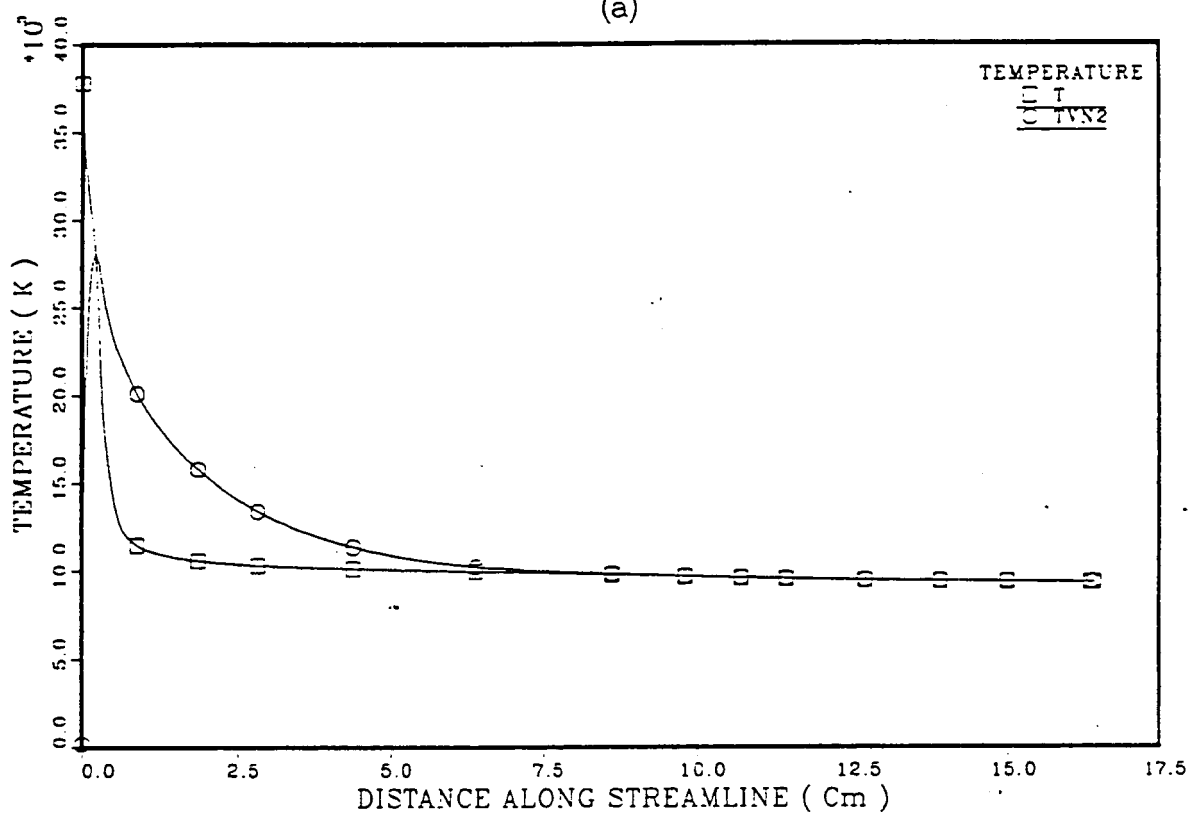


(b)

FIGURE 27. VEQ MODEL AT V=10 Km/s. RR3

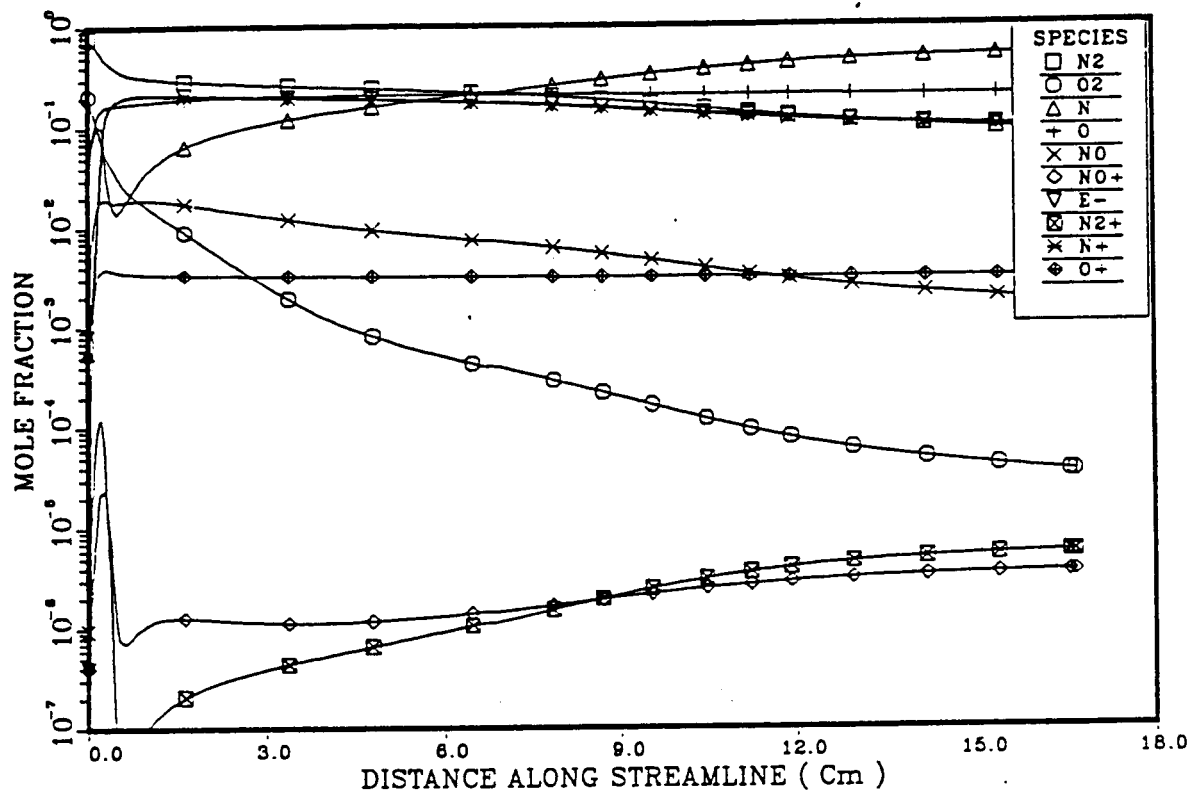


(a)

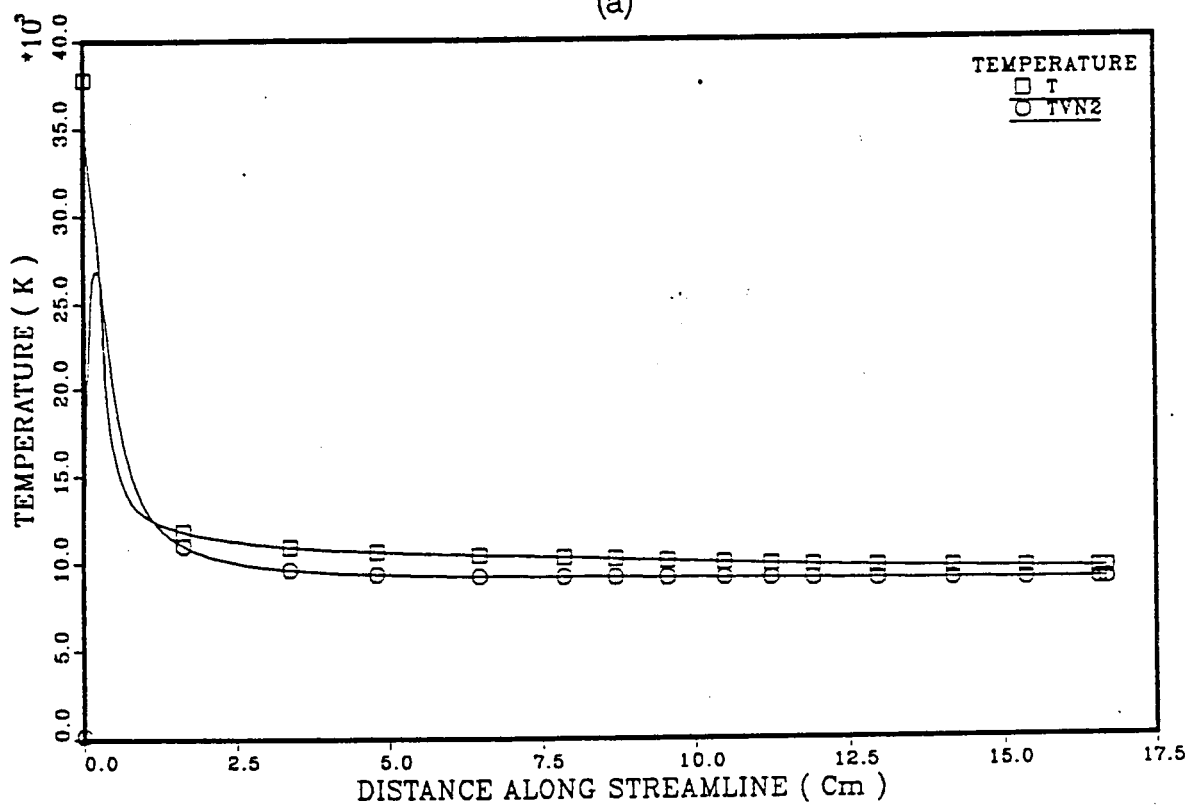


(b)

FIGURE 28. CVD MODEL AT $V=10$ Km/s, RR3

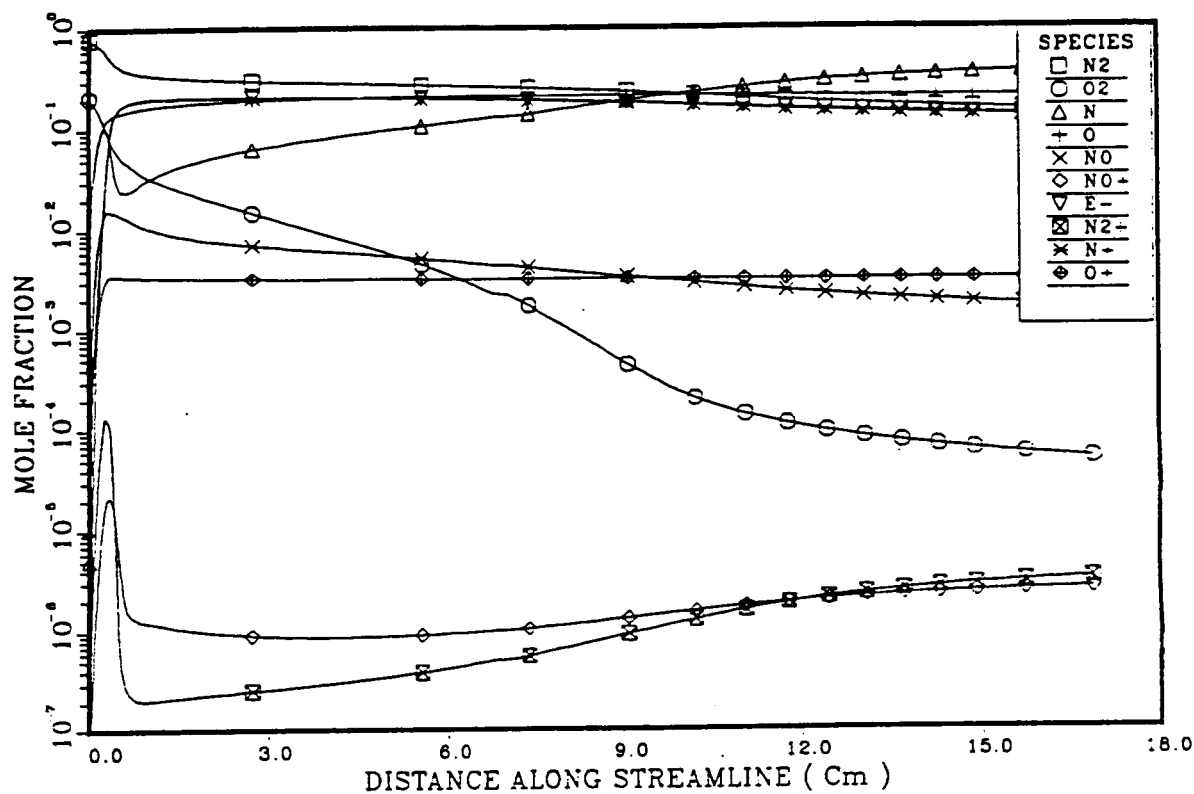


(a)

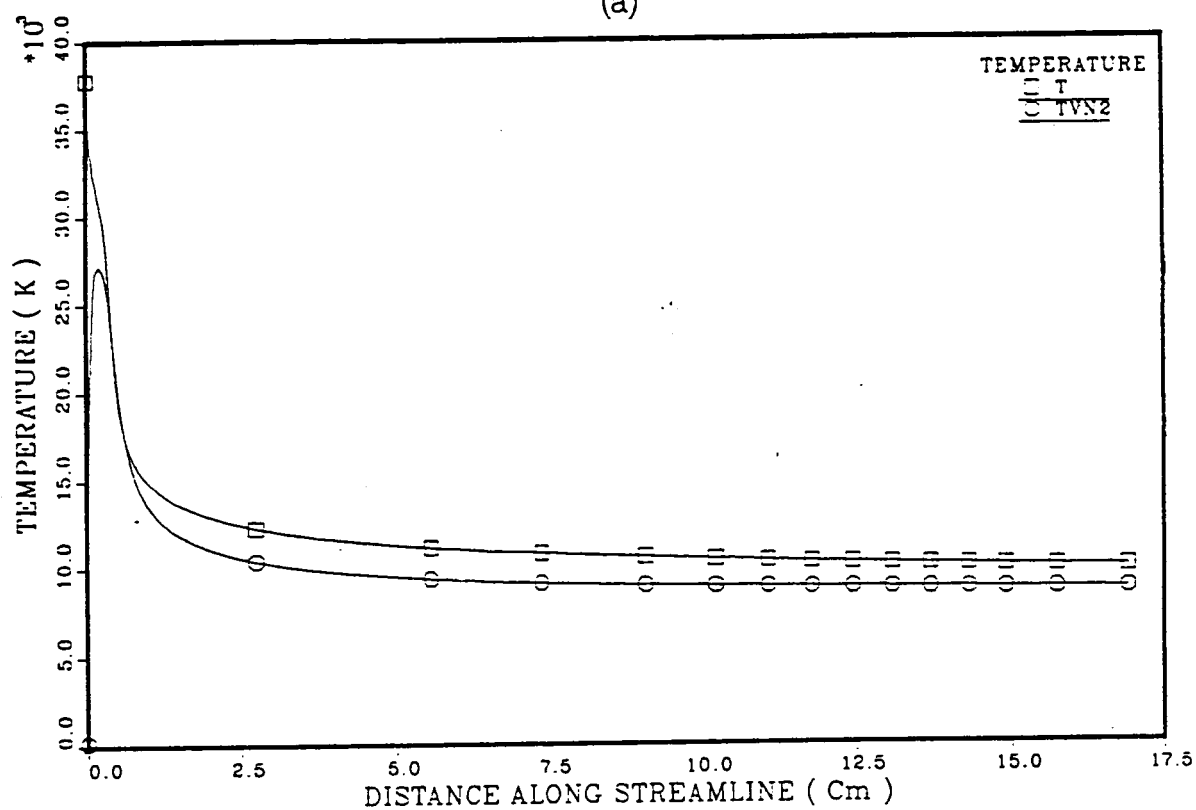


(b)

FIGURE 29. CVDV MODEL AT V=10 Km/s, RR3

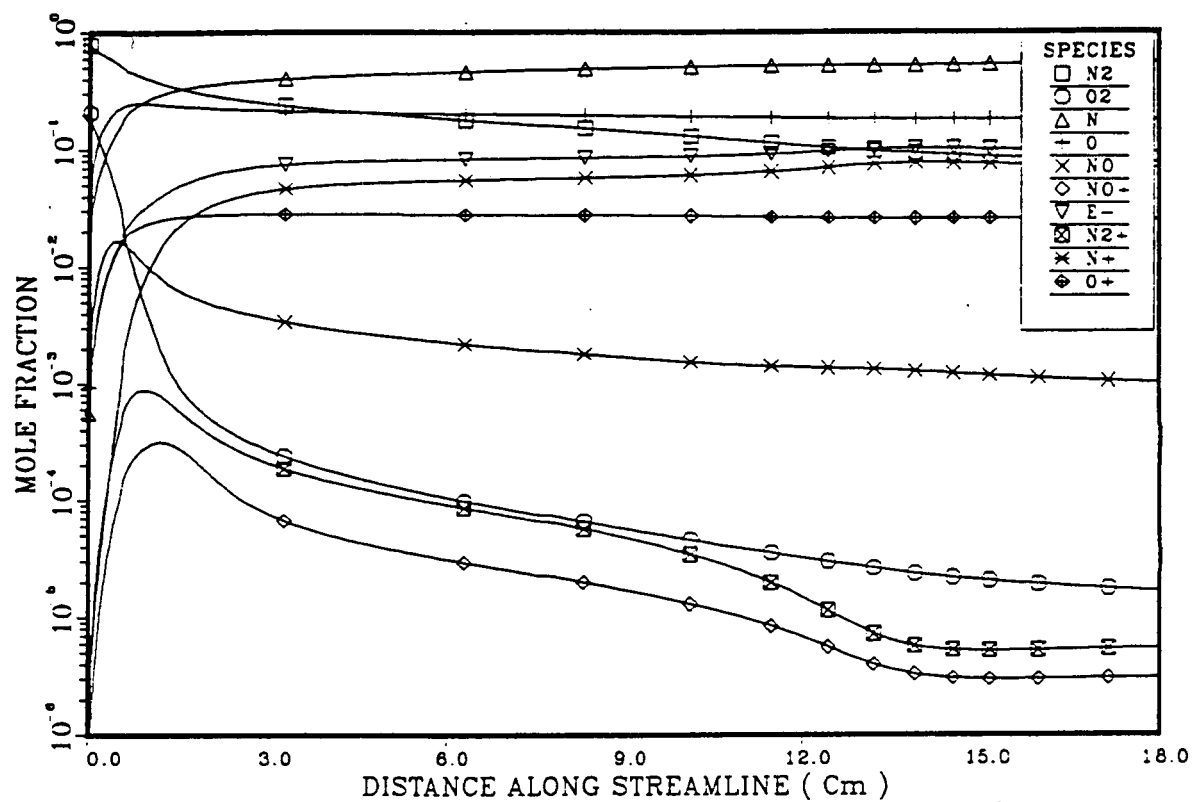


(a)

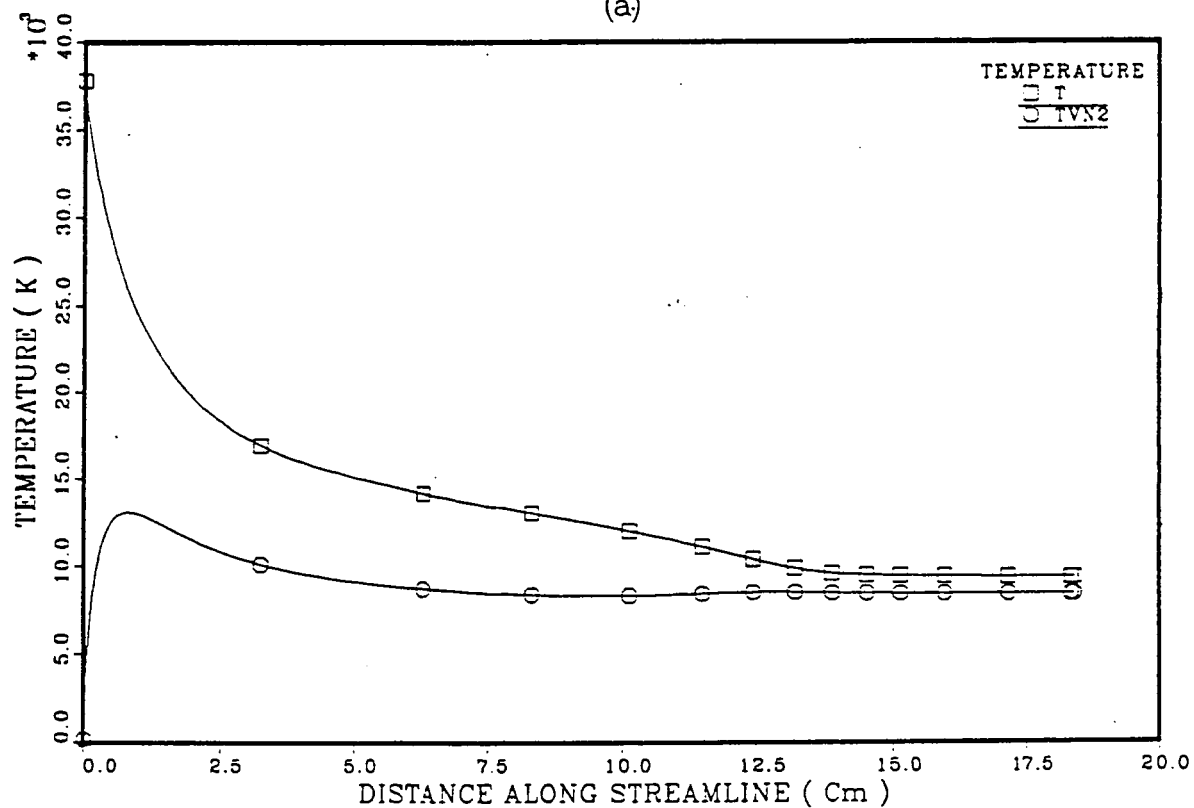


(b)

FIGURE 30. CVDV-P MODEL AT V=10 Km/s. RR3

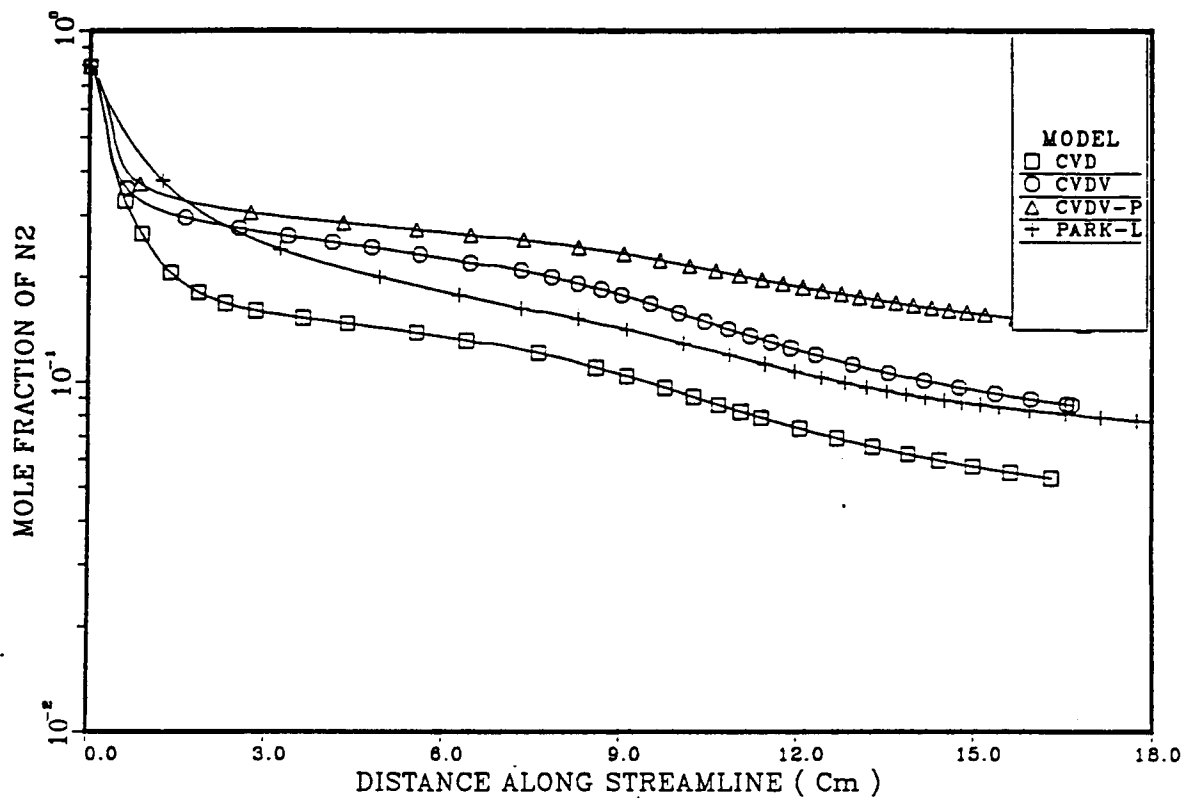


(a)

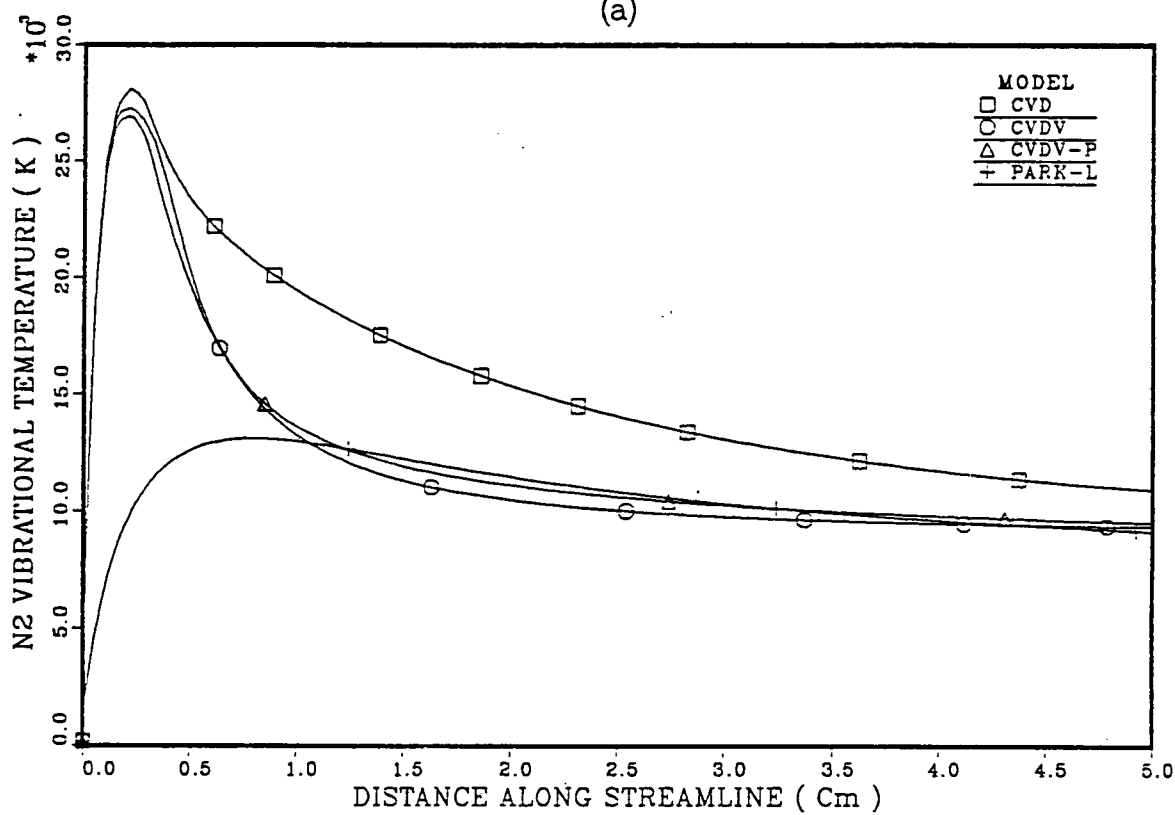


(b)

FIGURE 31. PARK-L MODEL AT V=10 Km/s, RR3

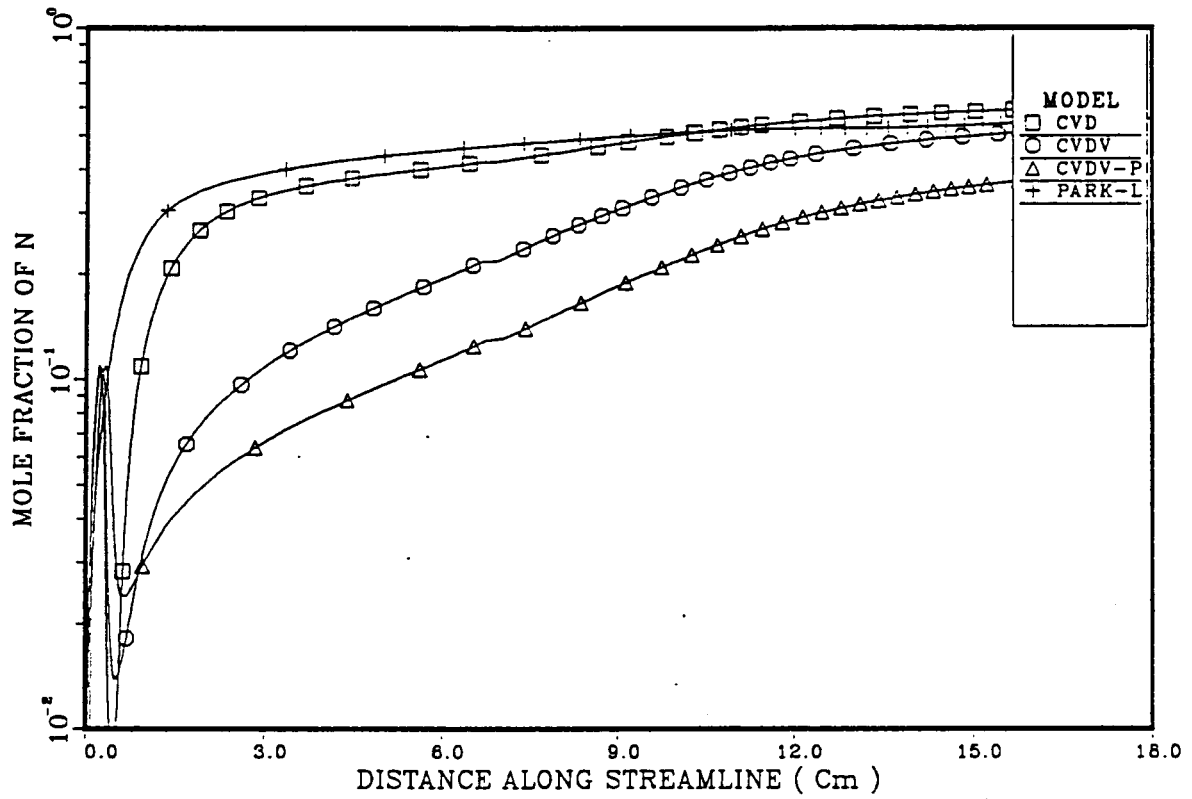


(a)

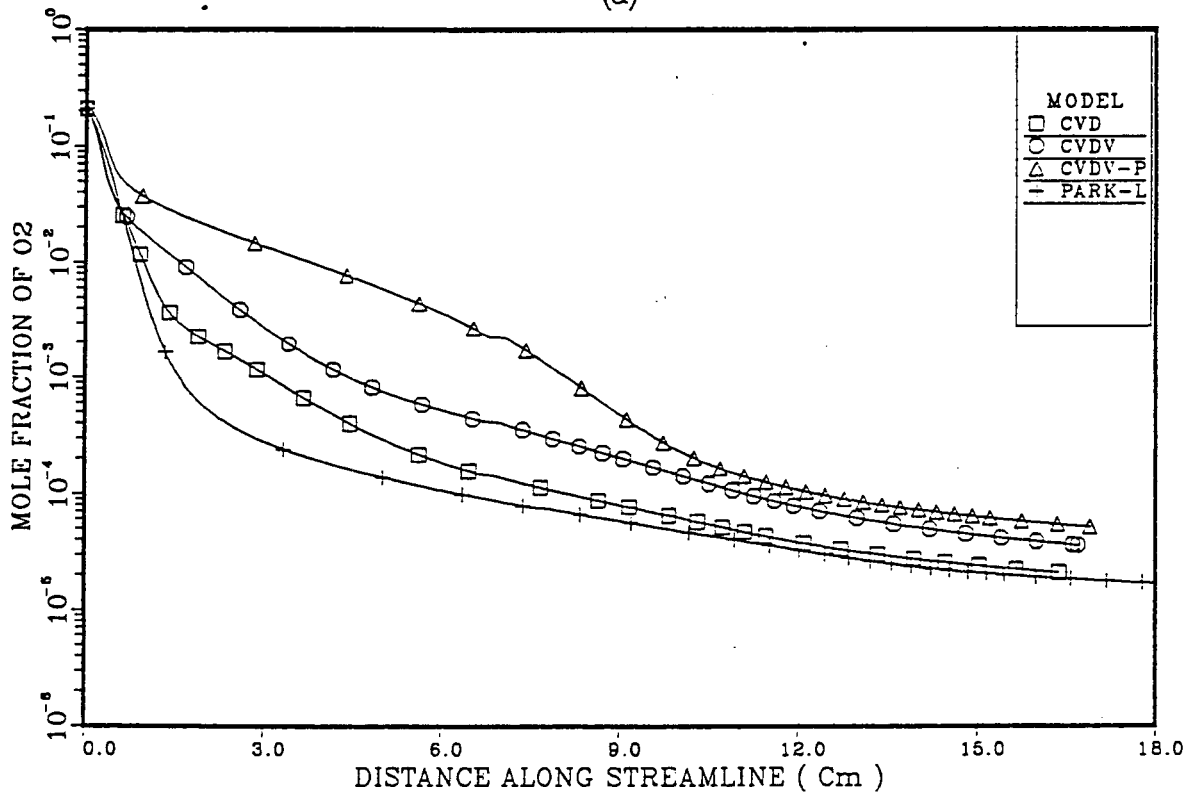


(b)

FIGURE 32. TVN2 AND N2 PROFILES AT V=10 Km/s, RR3

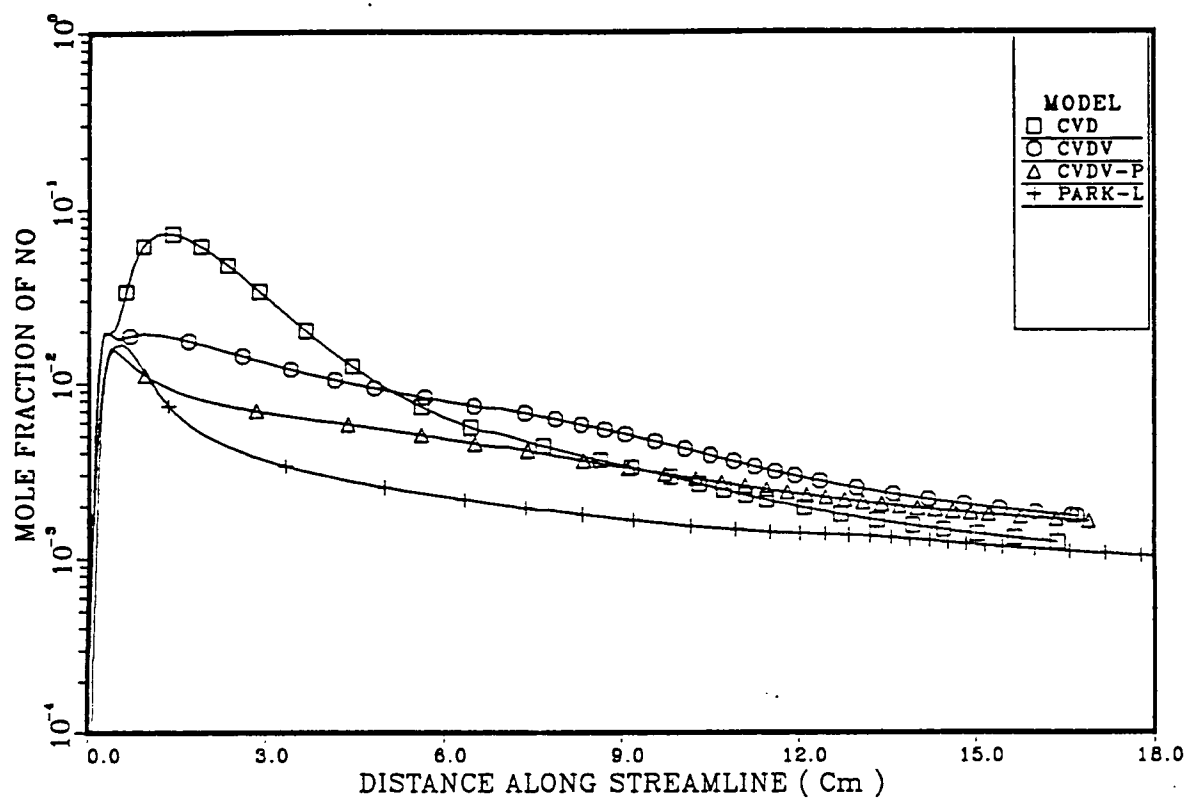


(a)

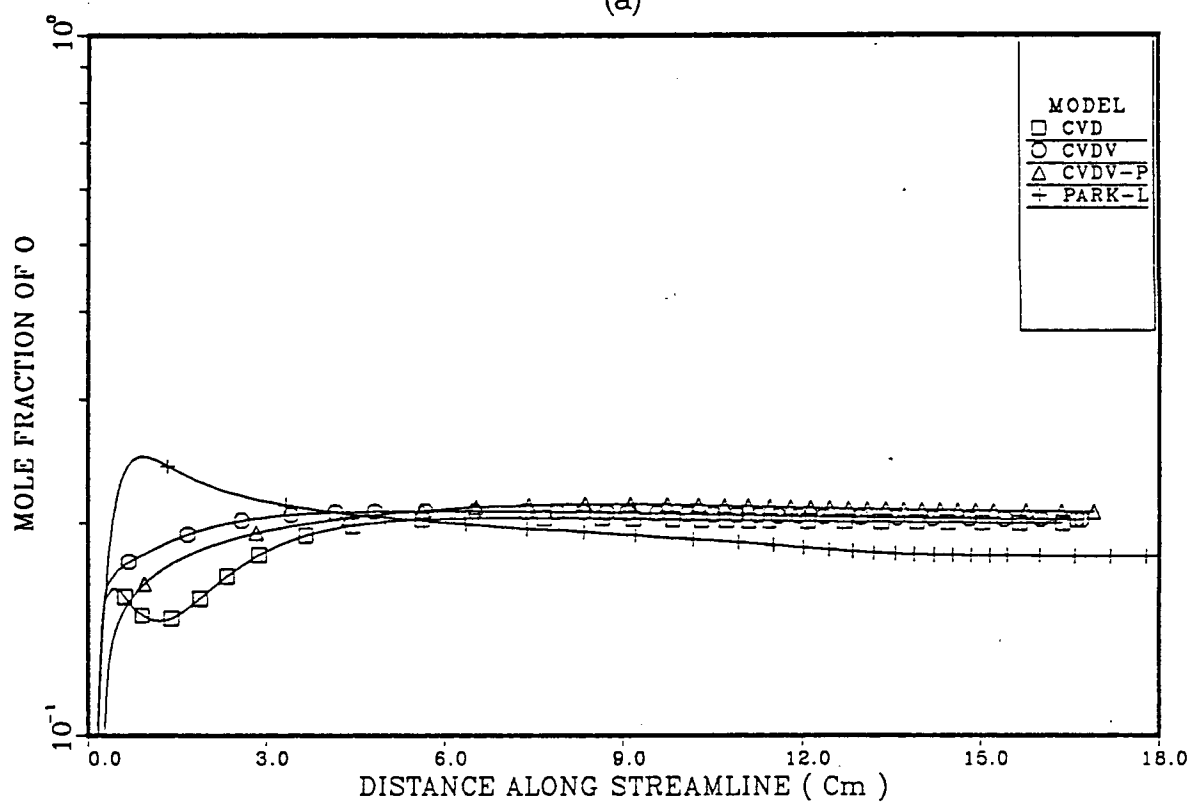


(b)

FIGURE 33. O2 AND N PROFILES AT V=10 Km/s, RR3

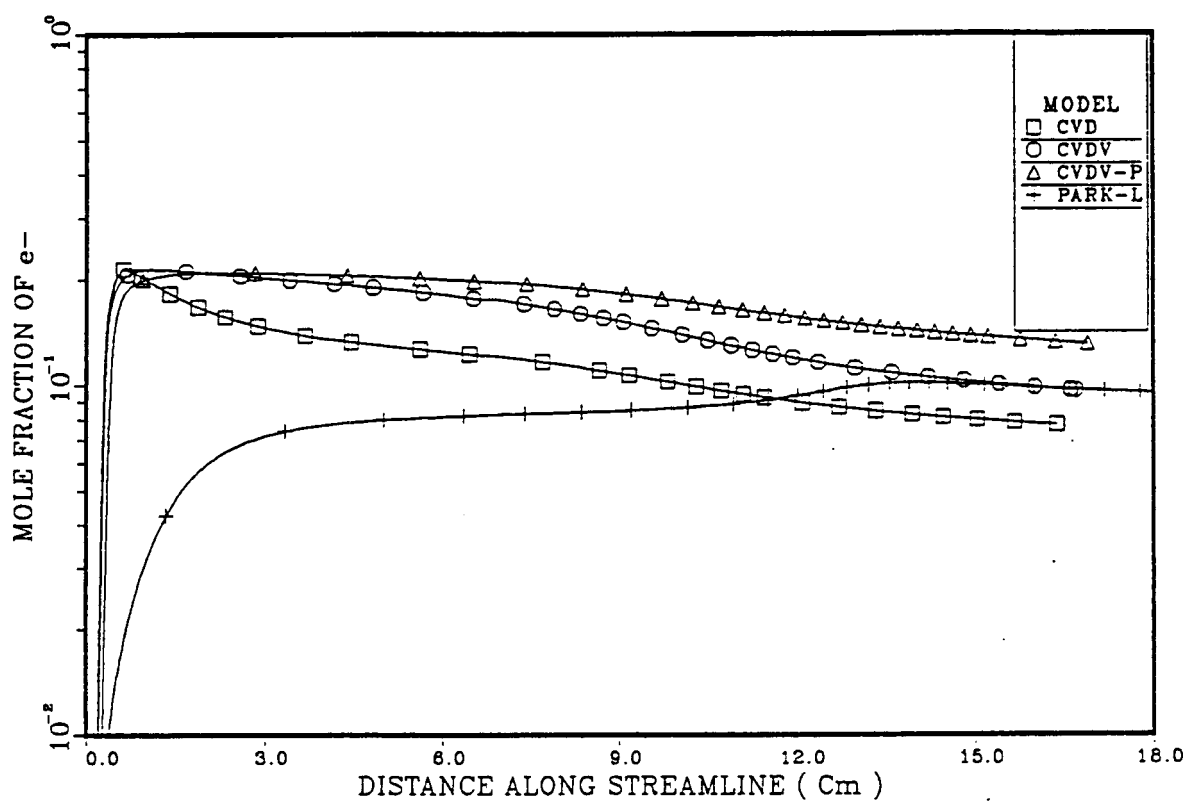


(a)

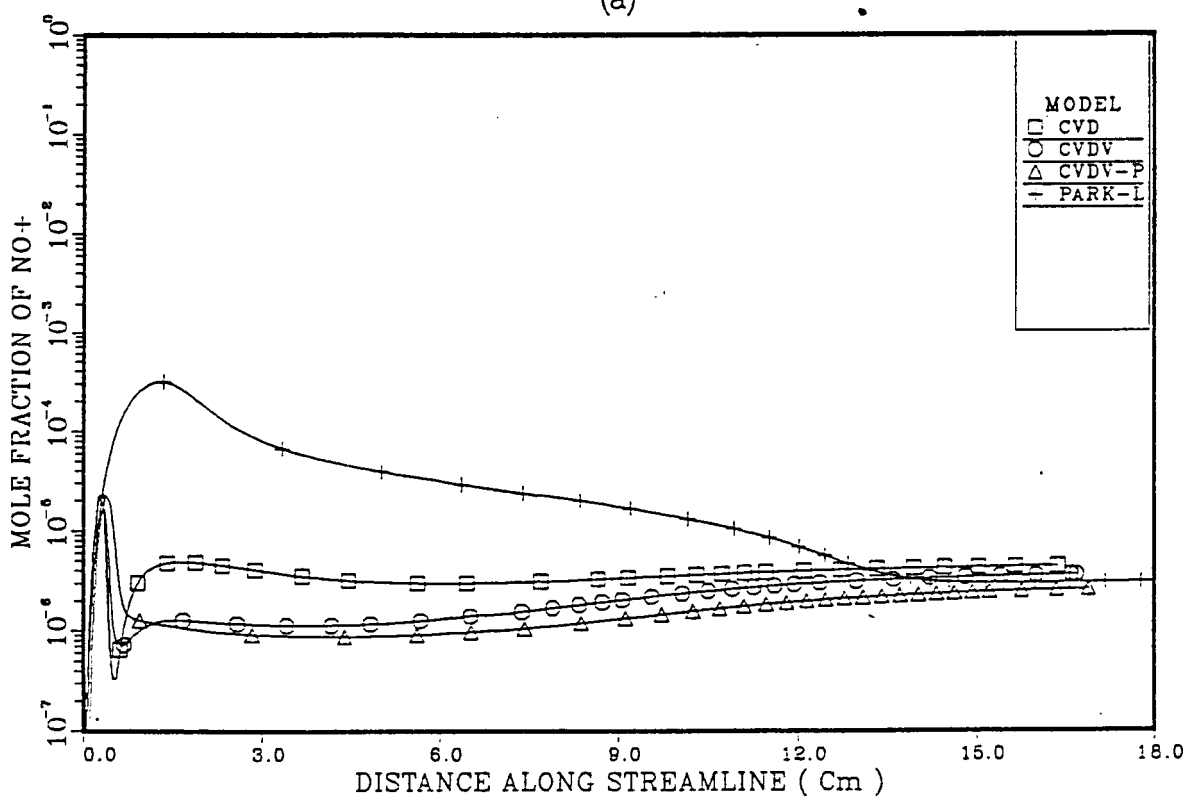


(b)

FIGURE 34. O AND NO PROFILES AT V=10 Km/s, RR3

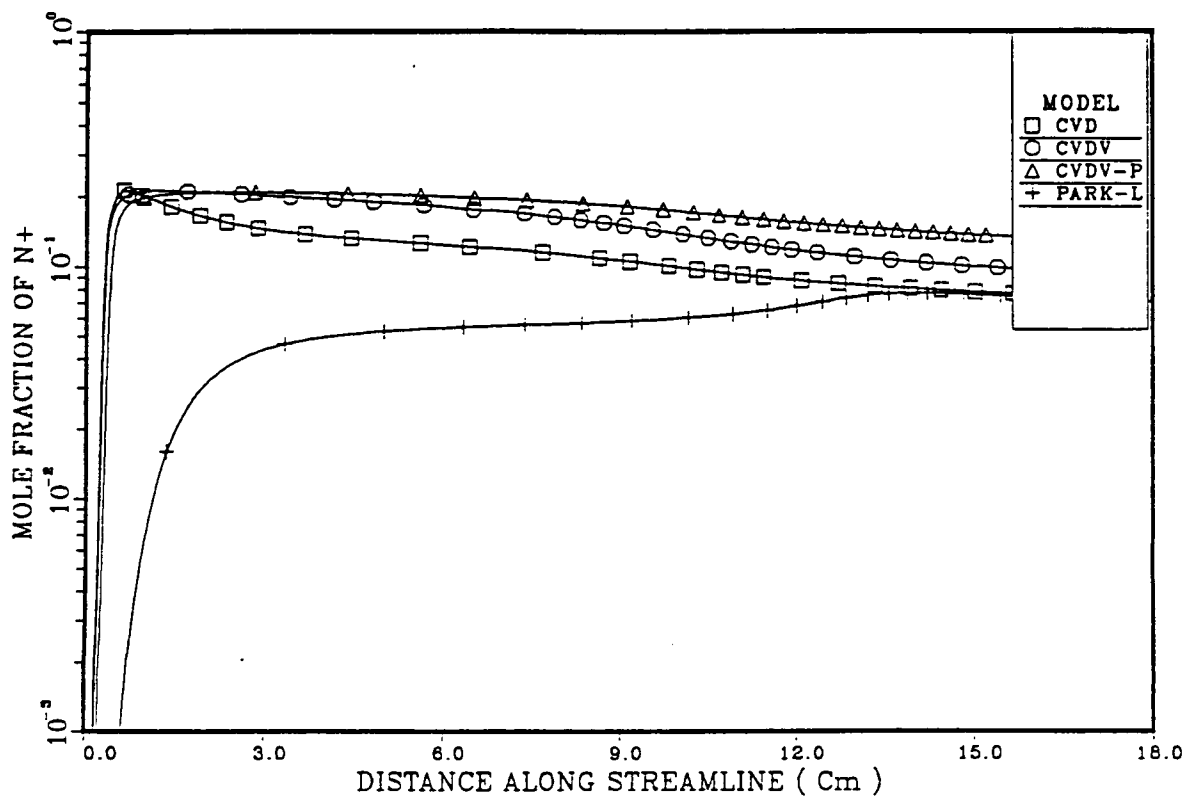


(a)

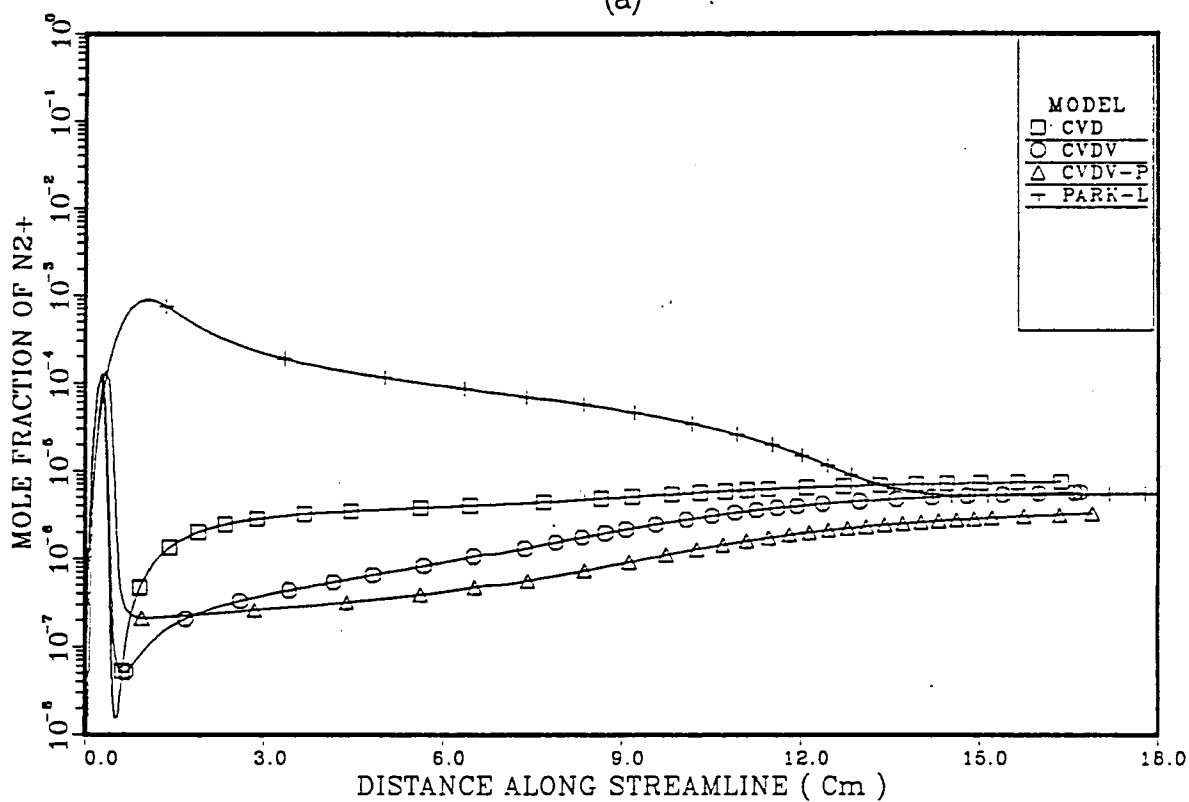


(b)

FIGURE 35. NO^+ AND e^- PROFILES AT $V=10$ Km/s, RR3



(a)



(b)

FIGURE 36. N_2^+ AND N^+ PROFILES AT $V=10$ Km/s, RR3

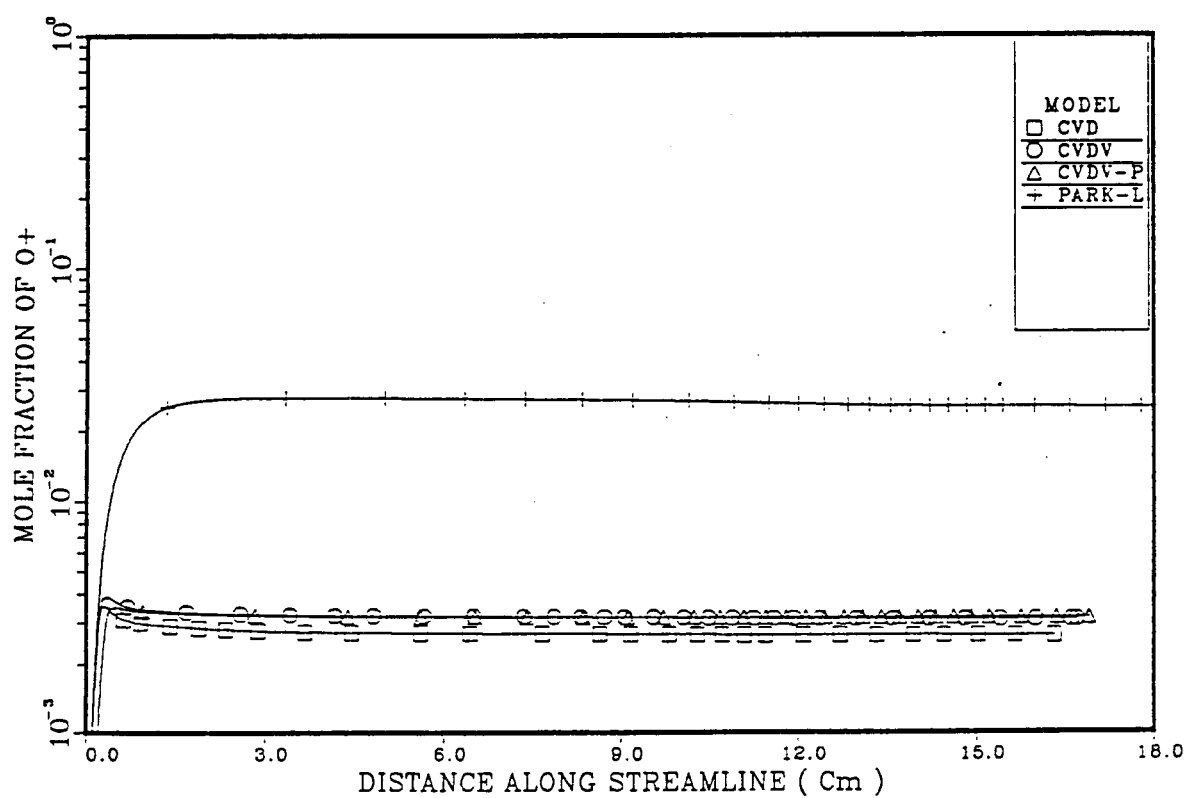


FIGURE 37. O+ PROFILE AT V=10 Km/s, RR3

immediately near the shock front. This ionization, in the vicinity of the shock front, is rapid enough to temporarily deplete the amount of nitrogen and reduce the N_2^+ and NO^+ concentrations several orders of magnitude. However, as the streamline distance increases, the ionization of N stabilizes enough to allow an increase in the concentrations of N, N_2^+ , and NO^+ . This phenomenon of electron avalanche is the dominant feature in the (RR3) results and is responsible for many unusual trends in the species profiles.

The CVD model, portrayed in Figure 28, shows a rapid decrease in translational temperature near the shock front. In addition, the N_2 vibrational temperature overshoot is far more extensive than previous CVD model results, which intuitively seems incorrect. This increase in vibrational temperature, coupled with the electron avalanche effect, produces abrupt changes in N_2^+ and NO^+ productions. These rapid changes indicate early electron recombinations of N_2^+ and NO^+ into atomic nitrogen and oxygen. This early recombination is followed by a short period of rapid ion formation, and then, by a gradual approach to equilibrium. This type of abrupt change, due to forward and reverse processes, can also be seen to a lesser degree in the concentration profile of NO.

The CVDV model results, which are shown in Figure 29, also depict a rapid decrease in translational temperature due to the increase in electron production as previously discussed for the vibrational equilibrium and CVD models.

As expected, the translational-vibrational temperature equilibration rate is slower than that of the CVD model. The CVDV model also displays an uncharacteristic N_2 vibrational temperature overshoot which, in itself, indicates the magnitude of the electron avalanche effect and the subsequent cooling effect on the translational temperature. The concentration profiles for N_2^+ and NO^+ display the same changes due to ionization as observed for the CVD model. As shown in Figure 33(a), the rapid N^+ production rate again produces a depletion of N followed by a slower approach towards equilibrium as compared to CVD. This slower N production rate, coupled with a significant decrease in O_2 dissociation, lowers the peak NO concentration and leads to the smooth transition shown in Figure 34(a).

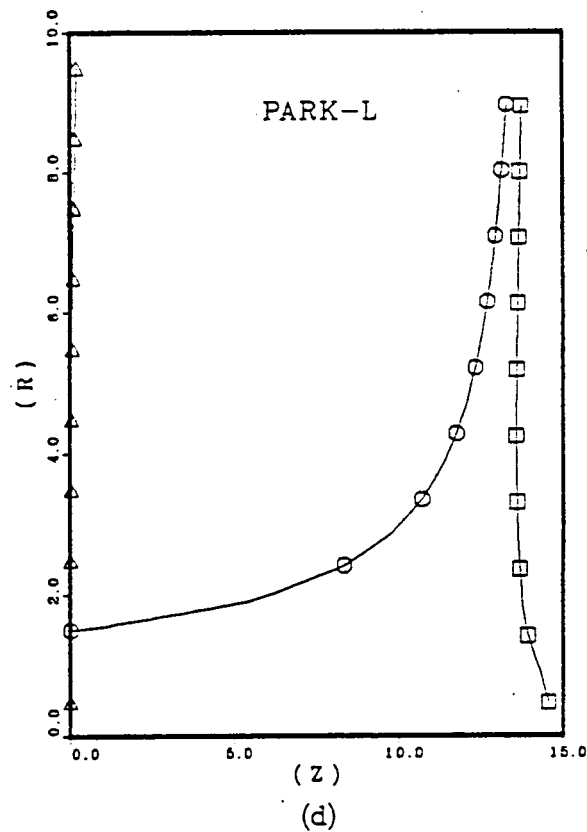
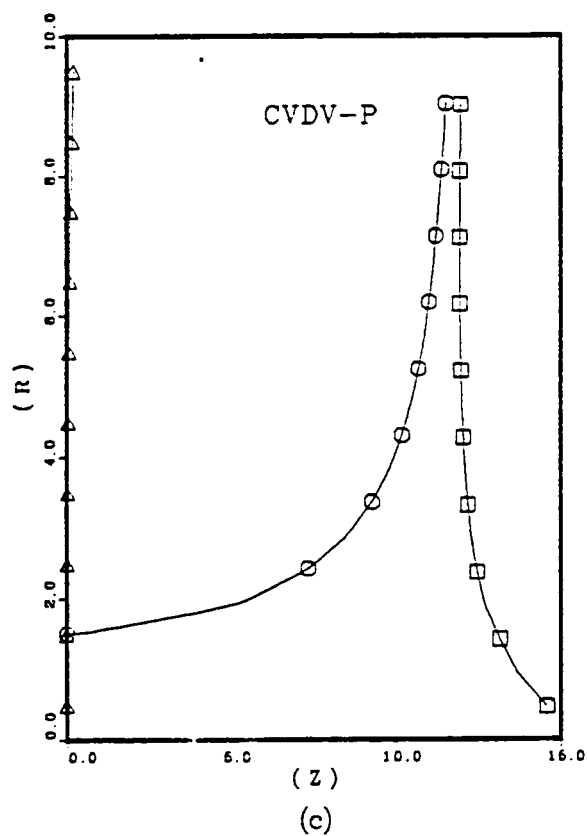
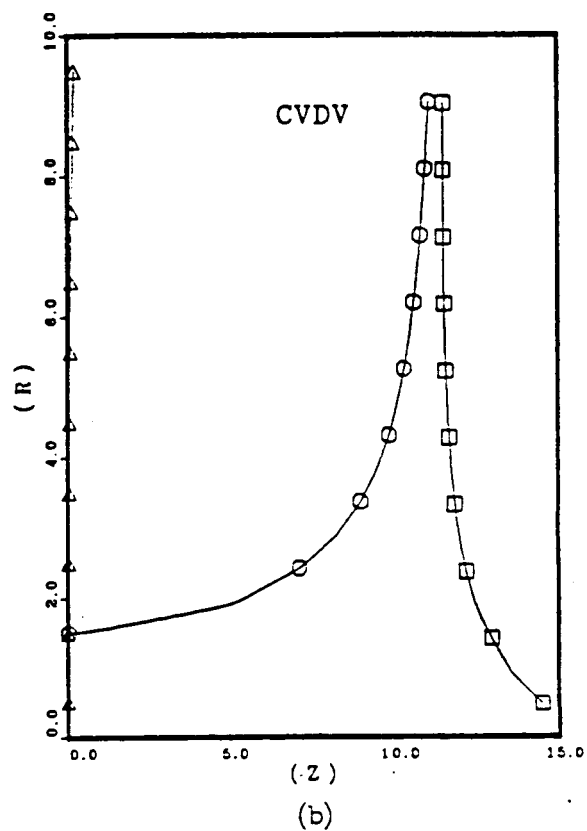
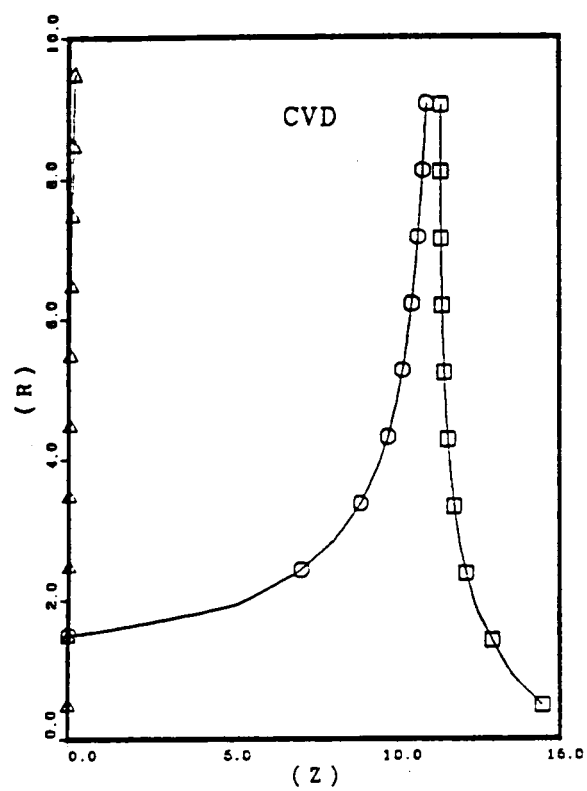
The CVDV-Preferential model (Figure 30) displays a decrease in the dissociation rates of N_2 and O_2 as compared to the CVDV model and are shown in Figures 32(a) and 33(b). Also, the concentration profile for the electrons [Figure 35(a)] shows a slight decrease in the initial rate of ionization due to electron avalanche, when compared to CVD and CVDV. Finally, the CVDV-Preferential temperature profiles, which are shown in Figure 30(b), do not depict any N_2 vibrational temperature overshoot of the translational temperature as was seen previously in the CVD and CVDV models.

As portrayed in Figure 31(b), the Park-Like model temperature profiles are similar to those obtained with the

other reaction rate models, but different than those obtained with the other vibration models. The modified Landau-Teller term within the vibrational rate equation prevents the N_2 vibrational temperature overshoot of the translational temperature, and lowers the peak vibrational temperature. Nevertheless, Figure 32(b) shows that along much of the streamline, the various coupling models closely predict the vibrational temperature of N_2 . As can be seen from the e^- and N^+ comparison profiles [Figures 35(a) and 36(a)], this decrease in peak N_2 vibrational temperature significantly decreases the electron avalanche effect due to its coupled T_{vN_2} rate controlling temperature. This delay in electron production prevents early deionization of ionized species near the shock front, also appears in the results for the other vibrational coupling models. Accordingly, the N_2^+ and NO^+ species profiles display a smooth overshoot in concentrations near the shock front. The Park-Like model, with this reaction rate set, also predicts a greater concentration of N^+ over that of O^+ . The aforementioned trend is uncharacteristic for this vibrational coupling model at this trajectory point. Finally, it should be noted that the concentration profiles of the Park-Like model do not follow the profiles associated with the CVDV-Preferential model. Apparently, the introduction of the rapid Kang and Dunn electron impact ionization rate, under N_2 vibrational temperature control, is the major reason for these differences in concentration profiles.

The shock, streamline, and body coordinates for this reaction rate set are displayed in Figure 38. All physical space coordinates maintained their basic shape and orientation when compared to the previous reaction rate model results. However, the CVD, CVDV, and CVDV-Preferential coupling models show significant changes in shock standoff distance. The standoff distance for (RR3) is actually 20 to 30 percent less for each model, except for the Park-Like model, which showed little change in standoff distance. The decrease in shock standoff distance is concurrent with the abrupt decrease in the translational temperature due to the rapid ionization as predicted by the CVD, CVDV, and CVDV-Preferential models. Due to this decrease in translational temperature, the density is higher and the detachment distance is smaller. On the other hand, for the Park-Like model, the translational temperature is higher along much of the streamline with a corresponding lower density and higher shock standoff distance.

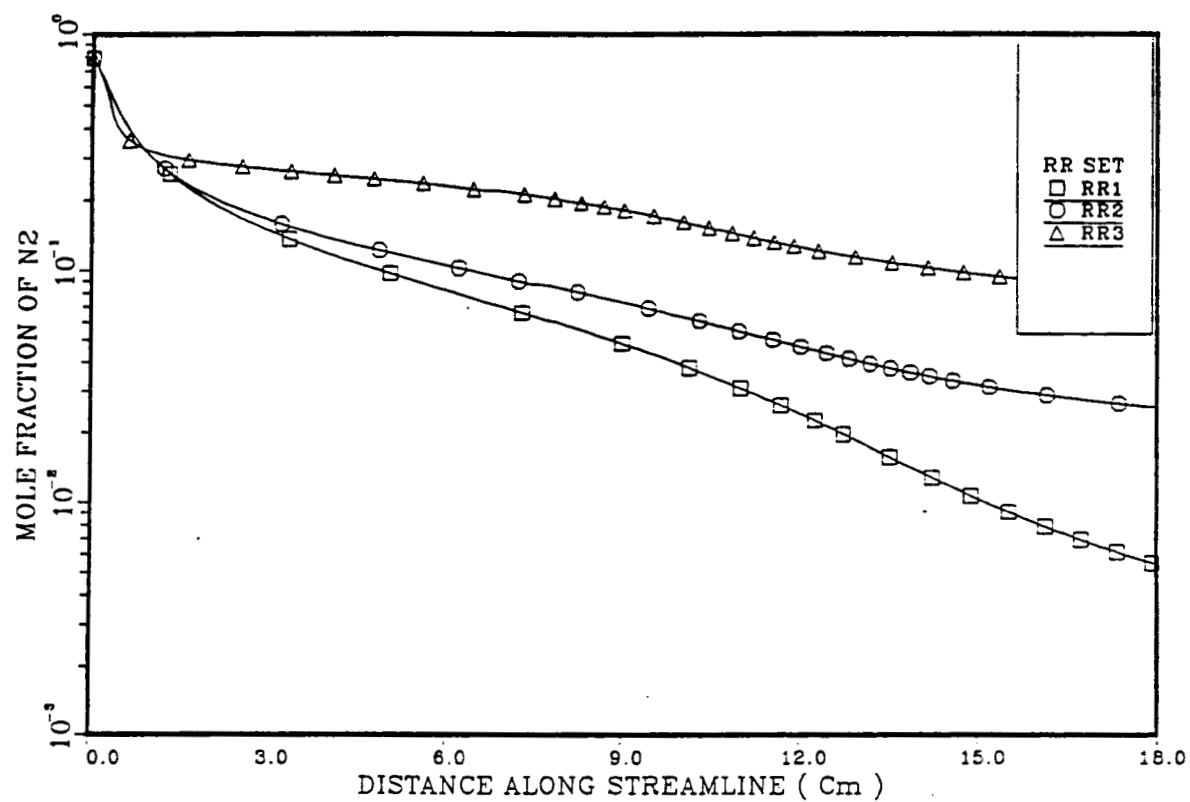
At the max-Q and exit trajectory points, temperature and concentration profiles have also been generated and studied for any significant trends. The results at the max-Q point show similar trends when compared to the results at the entry trajectory point. At the exit point, the most significant changes are the lack of a N_2 vibrational temperature overshoot for the CVDV model, and a greater concentration of O^+ over that of N^+ for the Park-Like model.

FIGURE 38. COORDINATES, $V=10$ Km/s, RR3

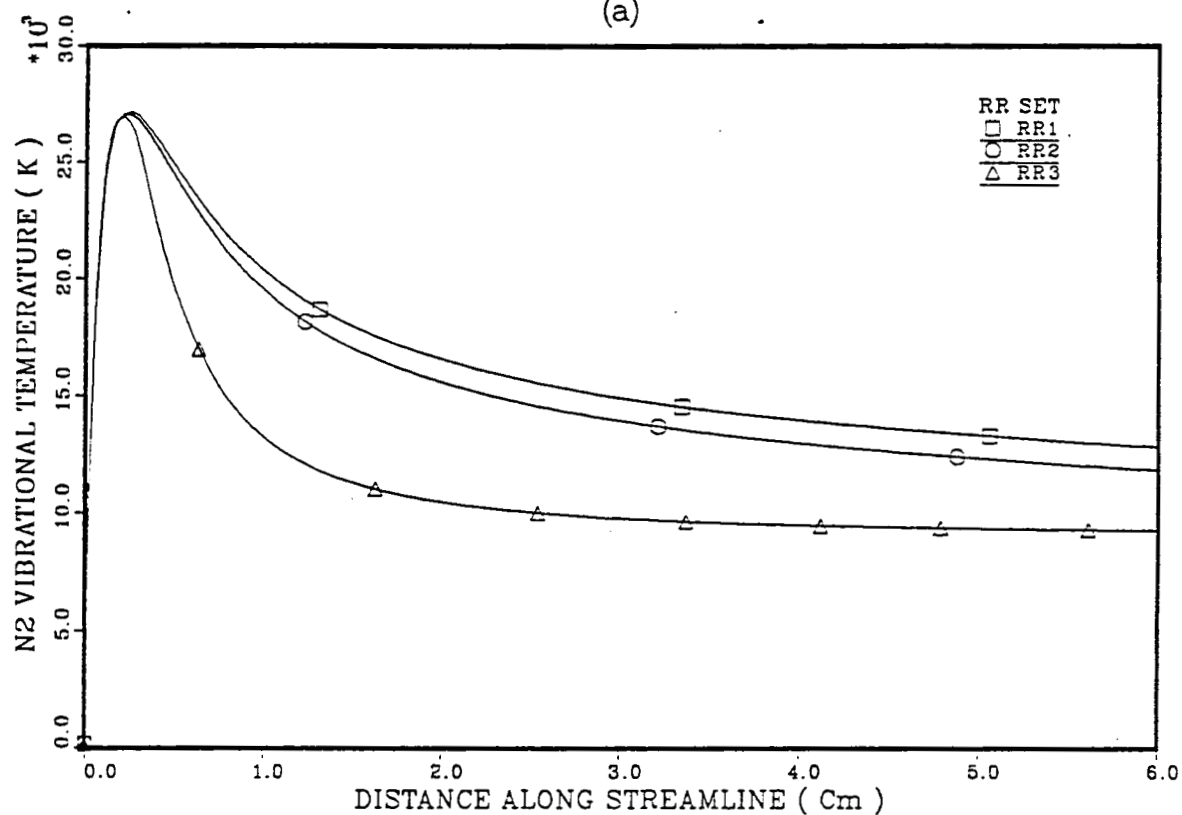
Reaction Rate Studies.

Although there has been previous discussion and comparisons of the effects of the reaction rate and vibrational coupling models, a detailed analysis has been performed on all reaction rate sets based on a single vibrational coupling model and trajectory point. However, this analysis has been limited to the CVDV and Park-Like models at all trajectory points in which only the significant trends will be discussed in interest of practicality.

The results for the CVDV model, at the entry trajectory point, are displayed in Figures 39-44. The changes in N_2 vibrational temperature [Figure 39(b)] depict that the first and second reaction rate models predict similar profiles but that the third model yields significantly lower vibrational and, in turn, electron temperatures. The influence of the additional ionized species and reactions are shown to be significant, in that the first and second sets display an order of magnitude difference in electron concentrations [Figure 42(a)]. Likewise, the influence of the electron impact ionization effect is also displayed by the (RR3) results for all concentration and temperature profiles. Surprisingly, the increase in ionization associated with the (RR3) electron impact reaction reduces the overall N_2 and O_2 dissociation rate while leading to an increase in the amount of NO concentration. This is possibly due to the immediate ionization of N and deionization of N_2^+ , NO^+ , and O^+ . The



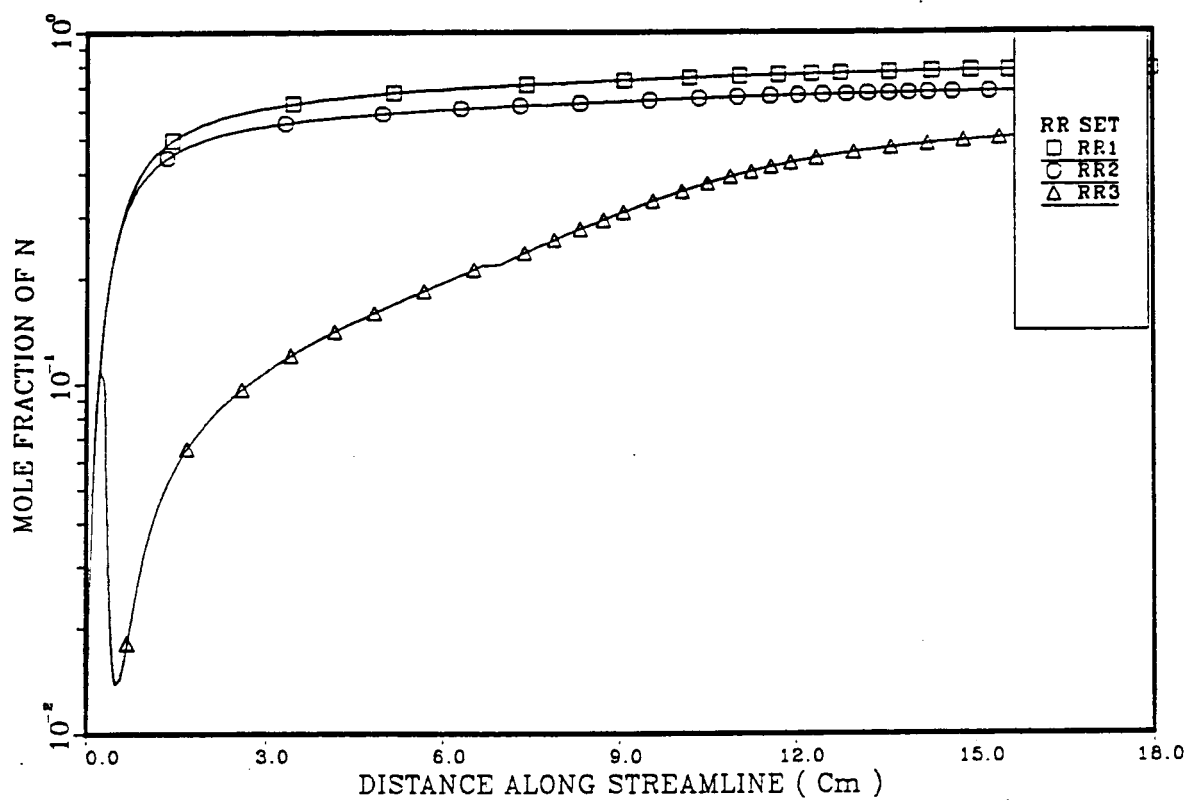
(a)



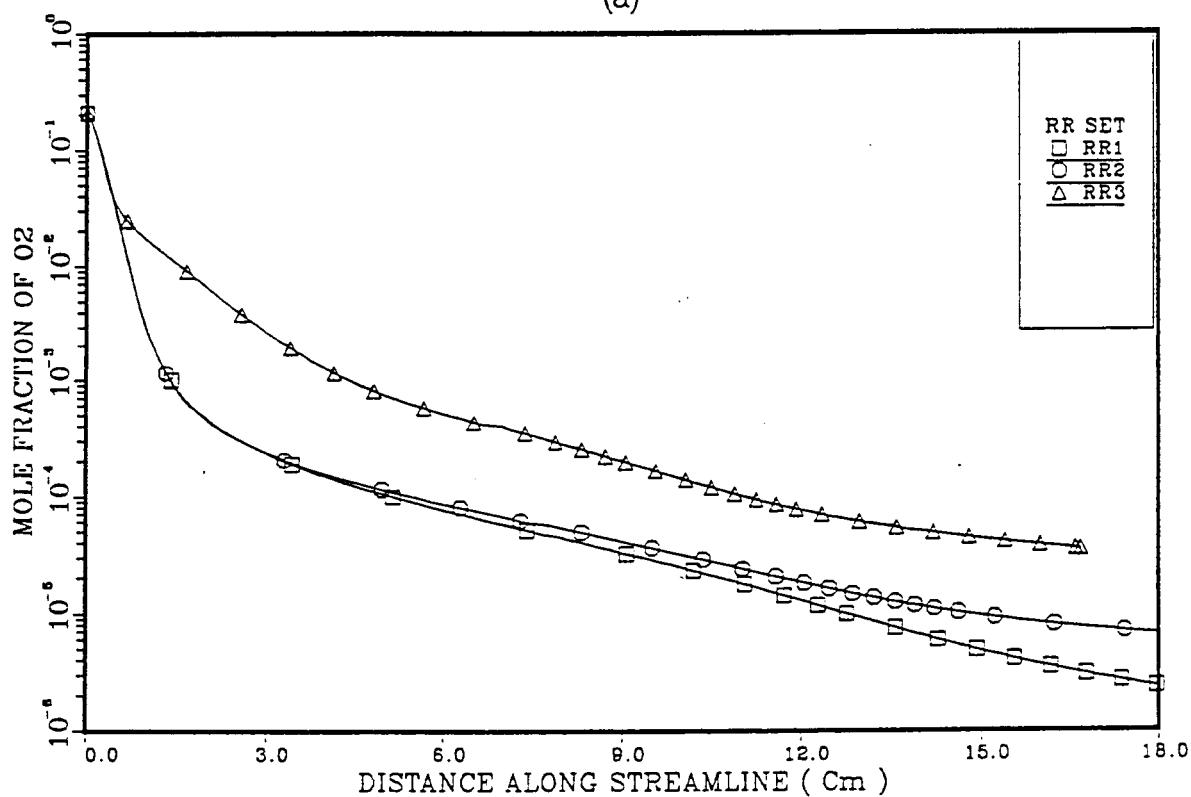
(b)

FIGURE 39. TVN2 AND N2 PROFILES AT V=10 Km/s, CVDV

144



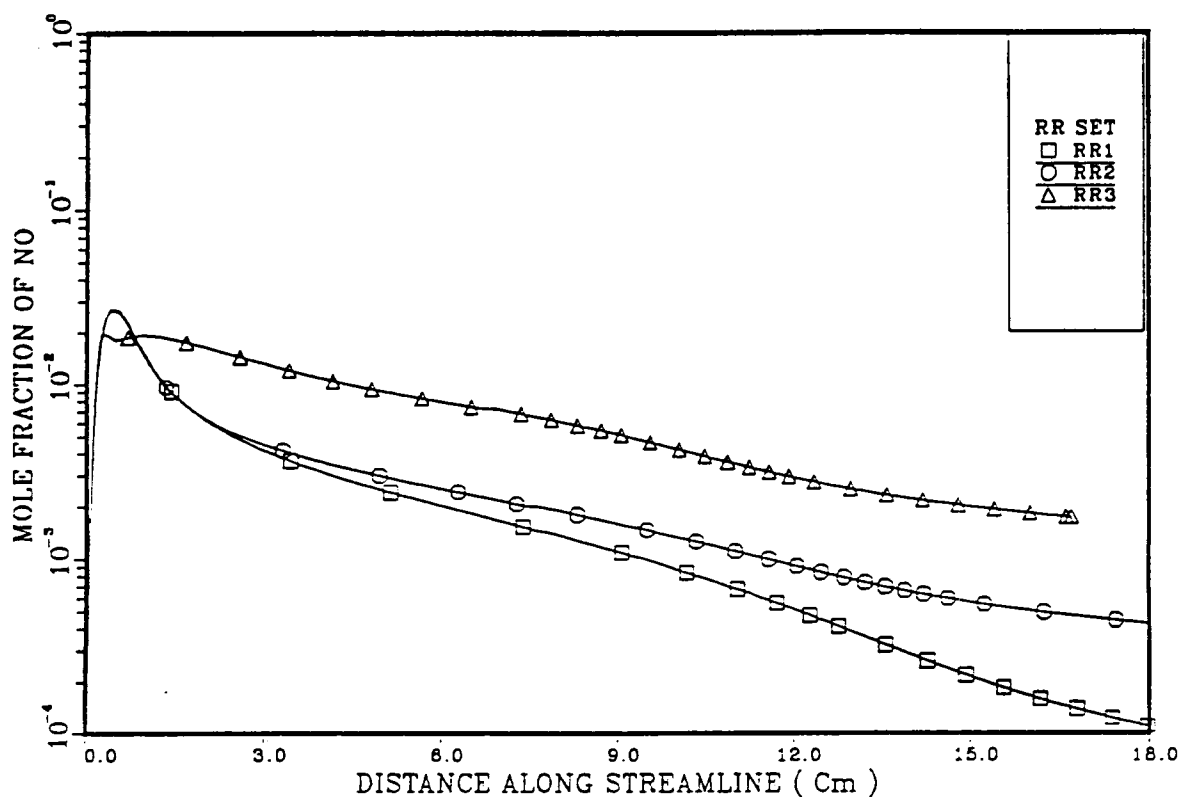
(a)



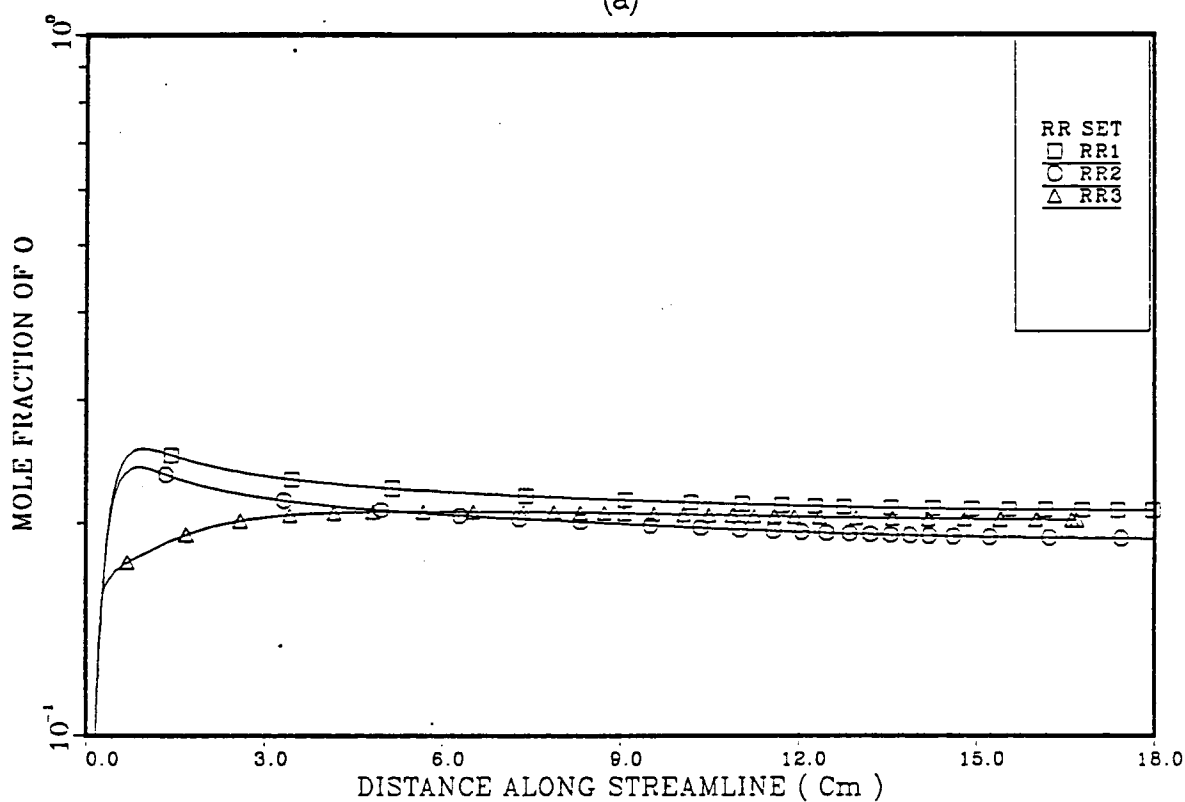
(b)

FIGURE 40. O₂ AND N PROFILES AT V=10 Km/s, CVDV

145

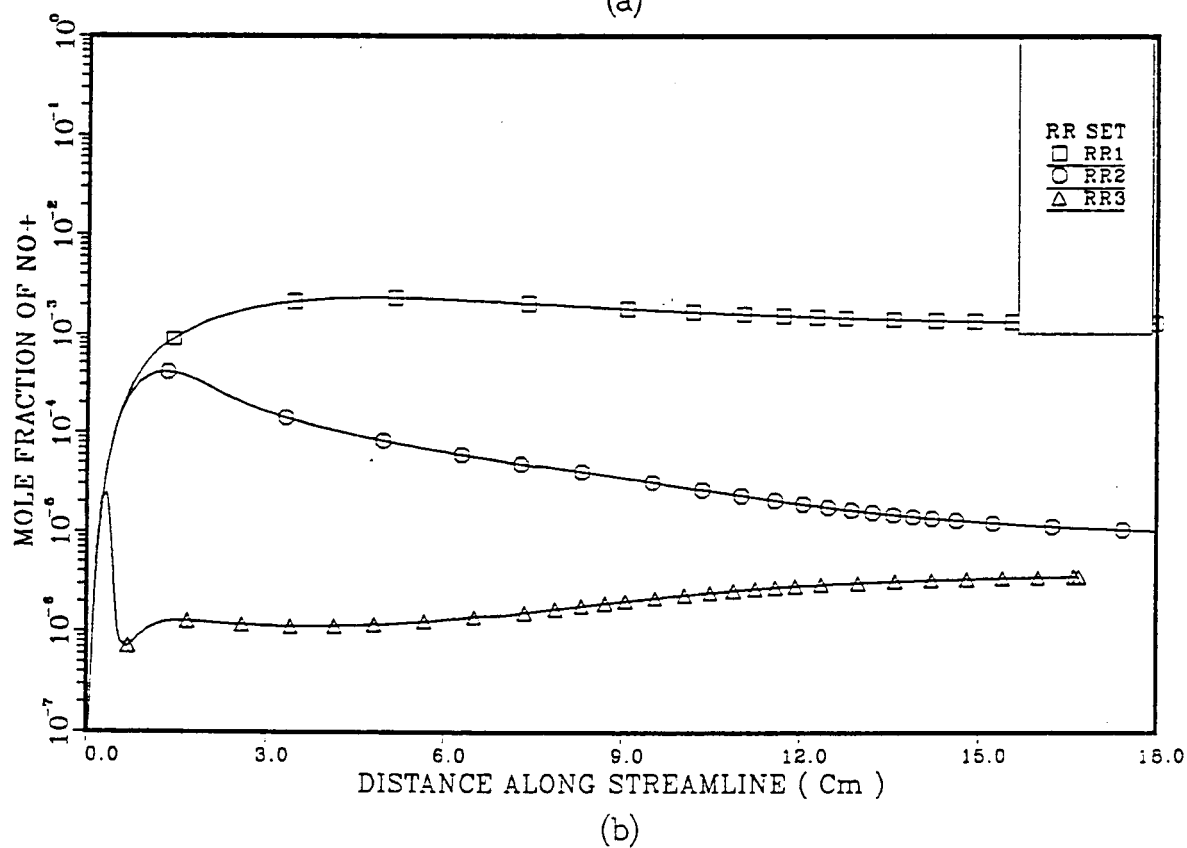
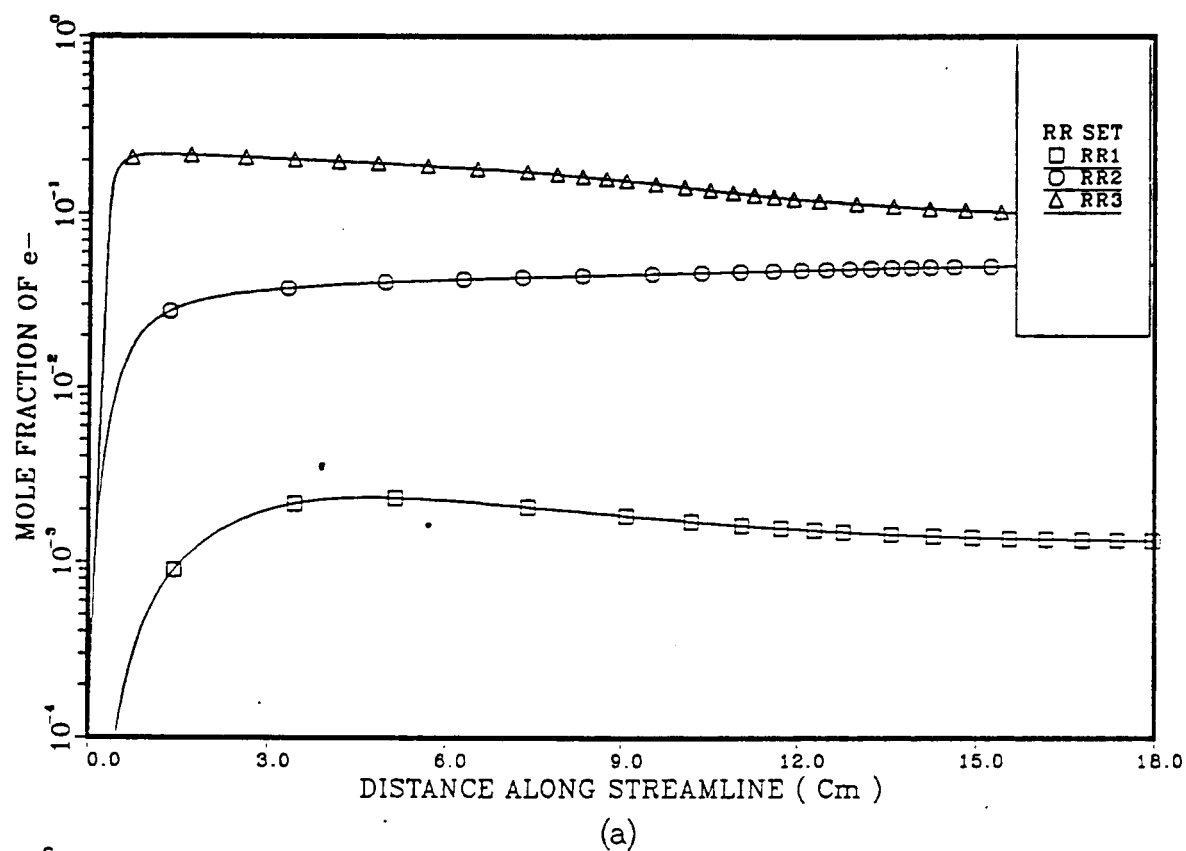


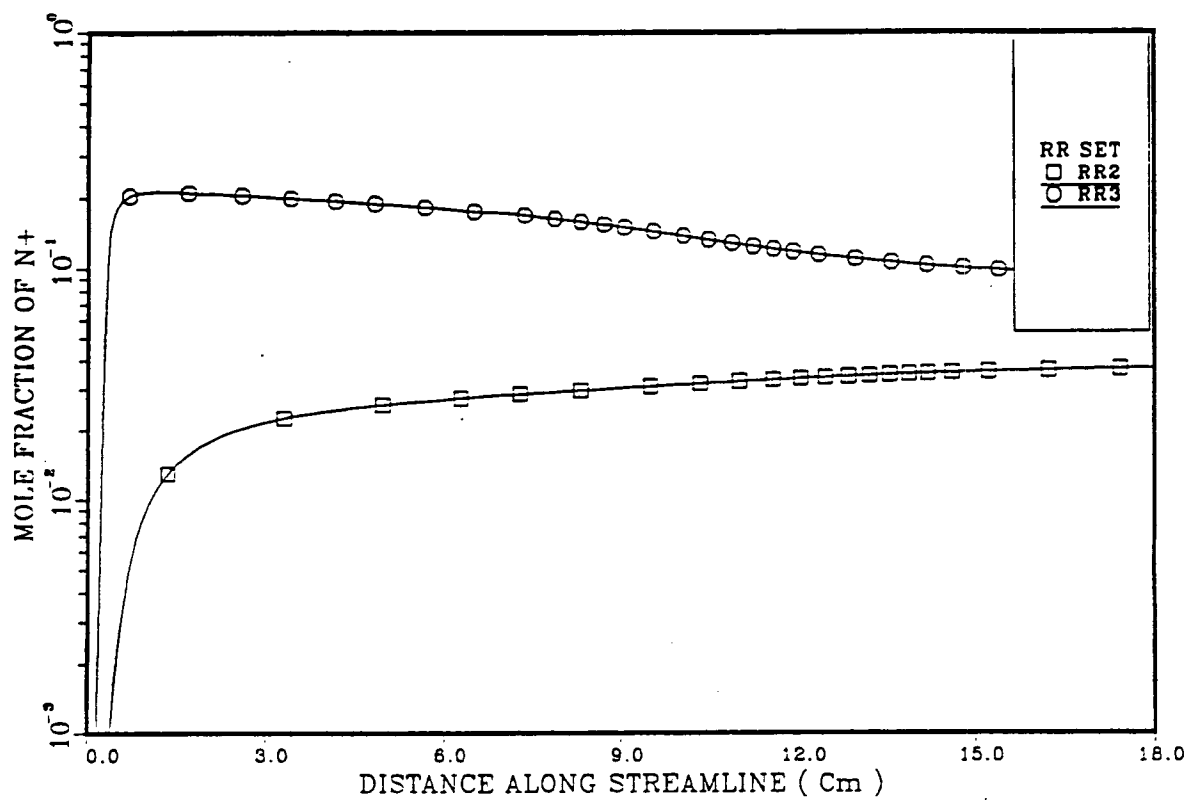
(a)



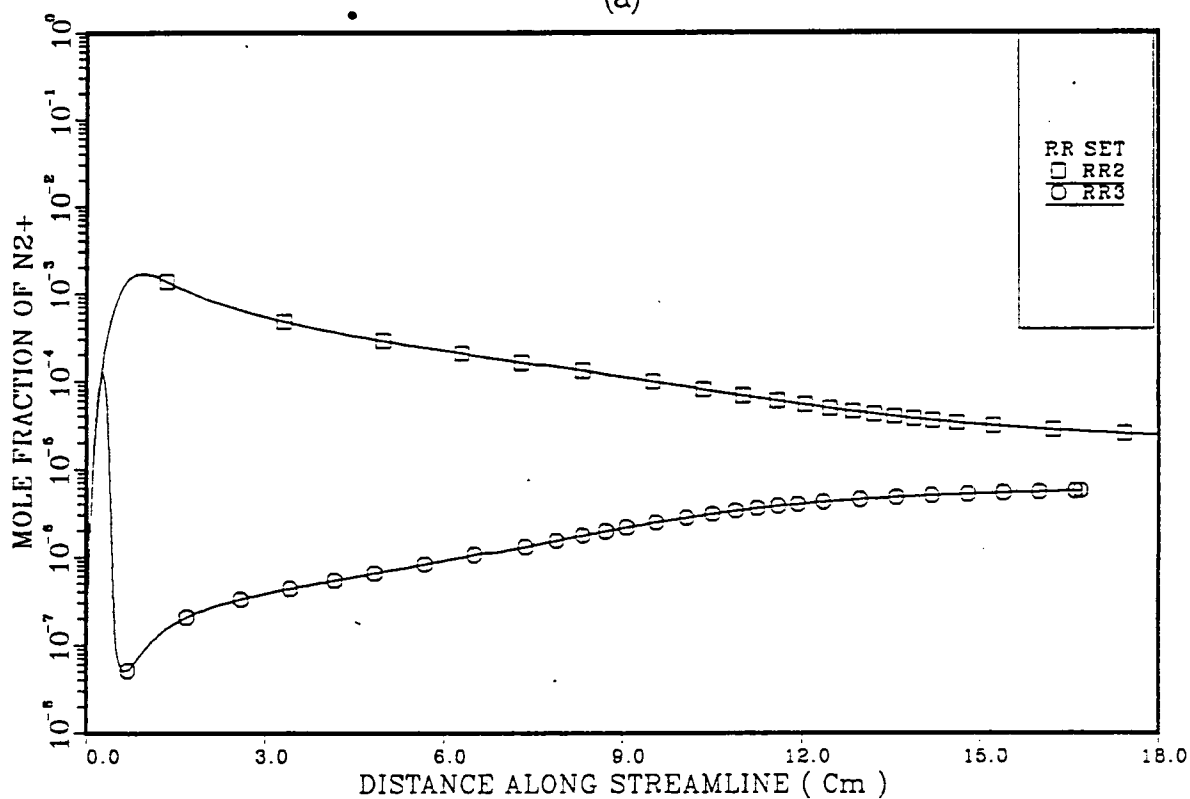
(b)

FIGURE 41. O AND NO PROFILES AT $V=10$ Km/s, CVDV

FIGURE 42. NO^+ AND e^- PROFILES AT $V=10$ Km/s, CVDV

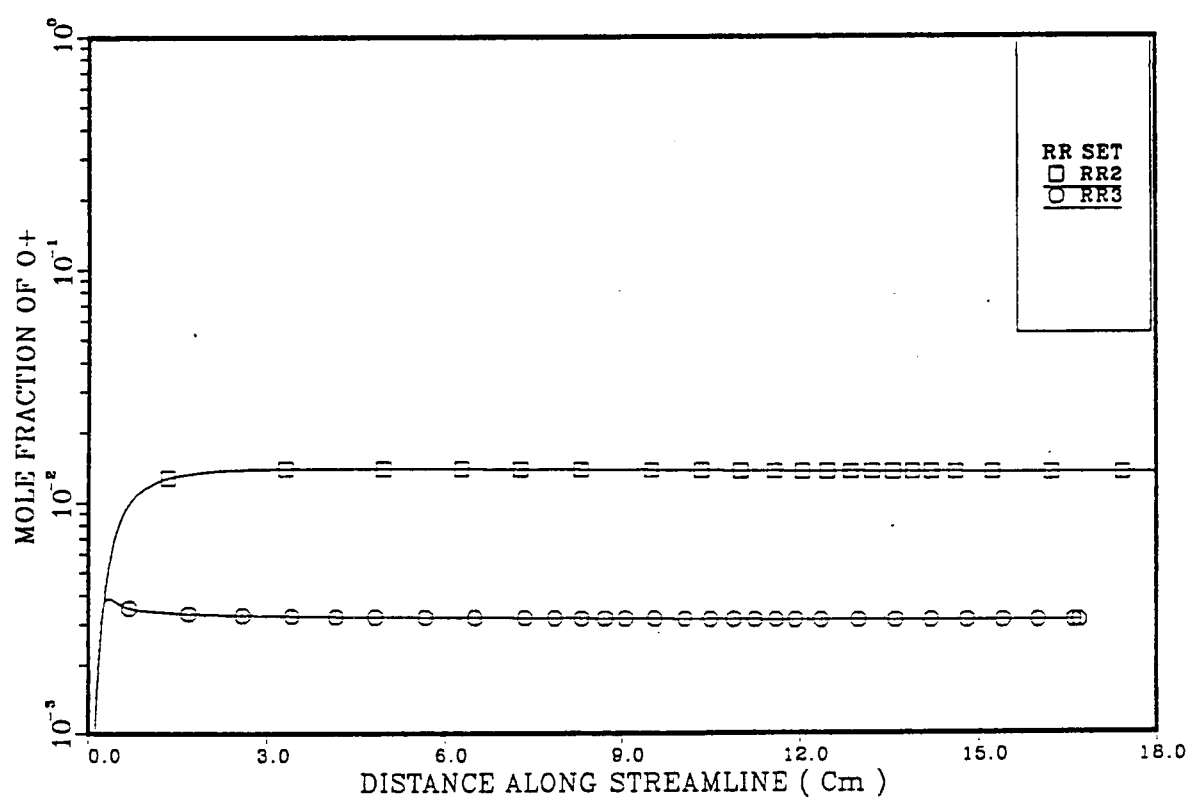


(a)



(b)

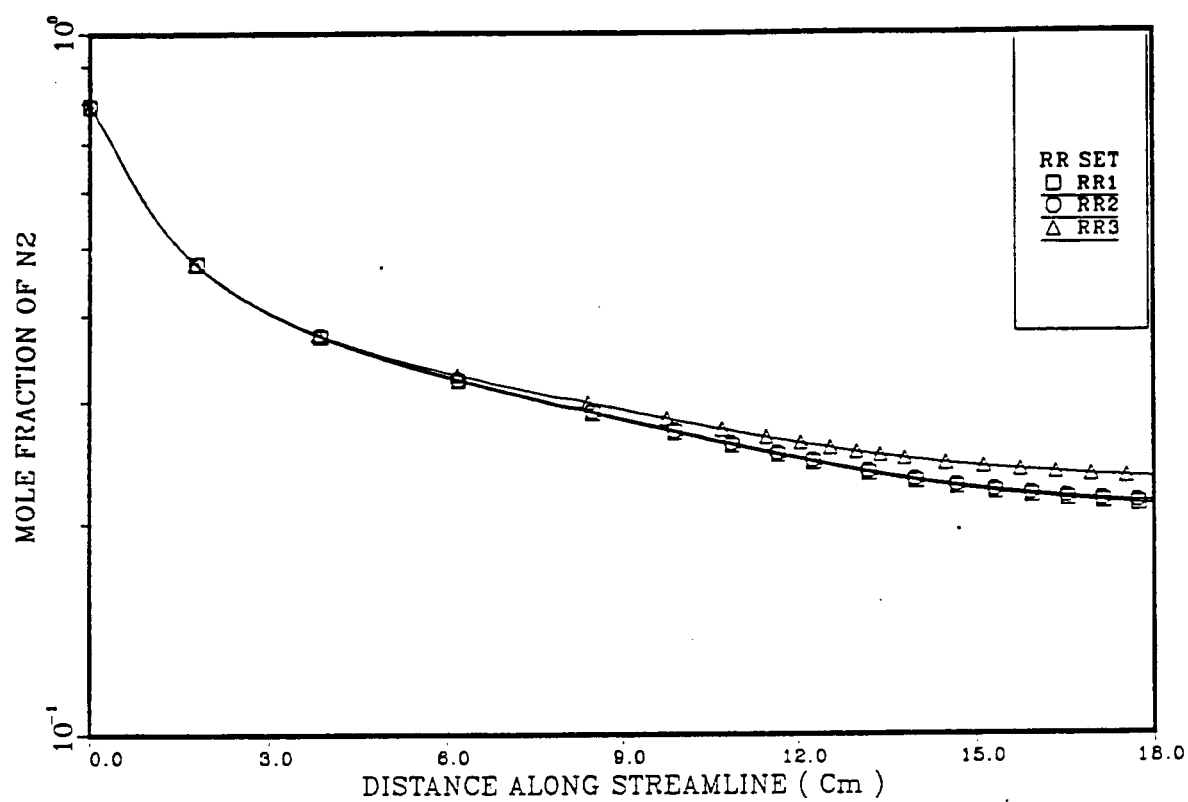
FIGURE 43. N_2^+ AND N^+ PROFILES AT $V=10$ Km/s, CVDV

FIGURE 44. O⁺ PROFILE AT V=10 Km/s, CVDV

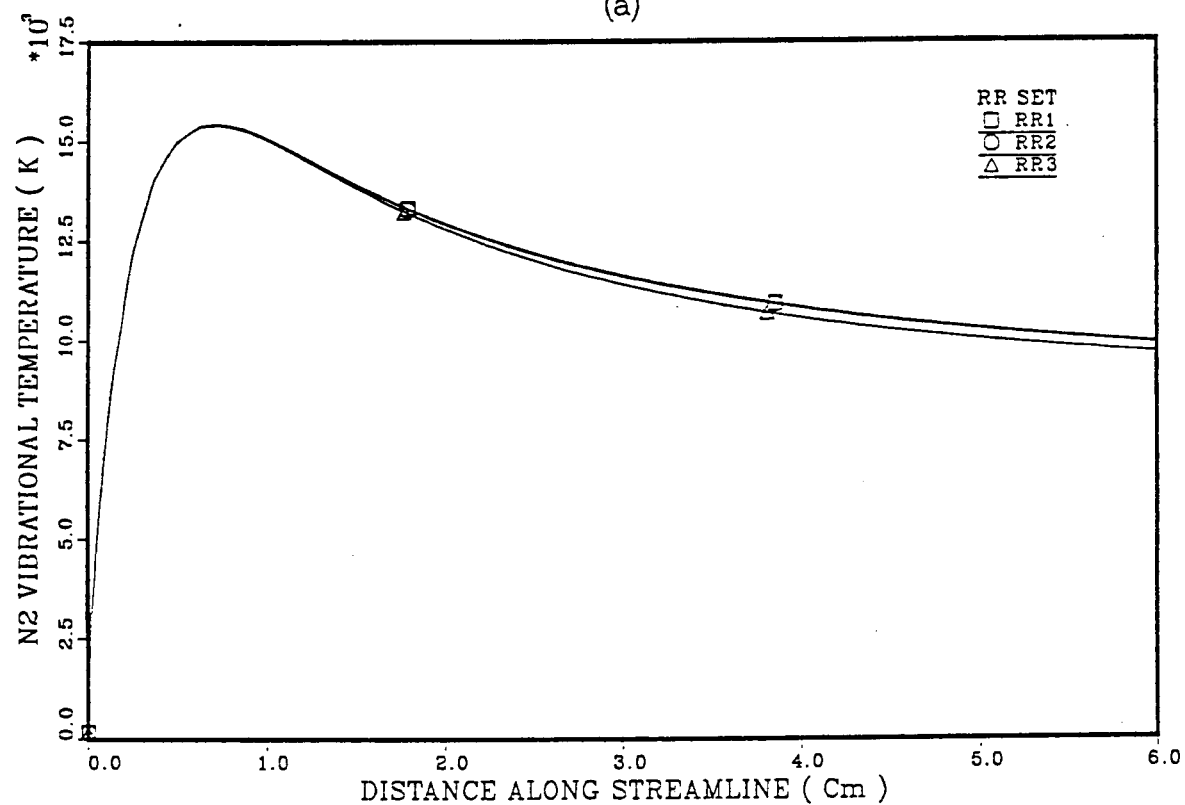
results for the max-Q trajectory point displayed similar profiles.

The exit point comparison results for the CVDV model are shown in Figures 45-50, and demonstrate that the concentration profiles for all ionized species are virtually identical for the three reaction rate model. These trends are extremely significant because they justify a considerable savings in computational time by using the smaller reaction rate model while still maintaining engineering accuracy.

Comparison temperature and concentration profiles have also been generated using the Park-Like model and all reaction rate sets. The Park-Like model results predict similar trends to the CVDV model for all three trajectory points. However, the Park-Like model displays better agreement between all reaction rate sets over a wider velocity range when compared to the CVDV model. Possibly, the use of the Park-Like model could justify the use of the smaller reaction rate set, at the higher velocities, in order to take advantage of reduced computational intensity. Of course, the larger reaction rate sets would have to be used in order to provide a more accurate analysis of the ionization processes.

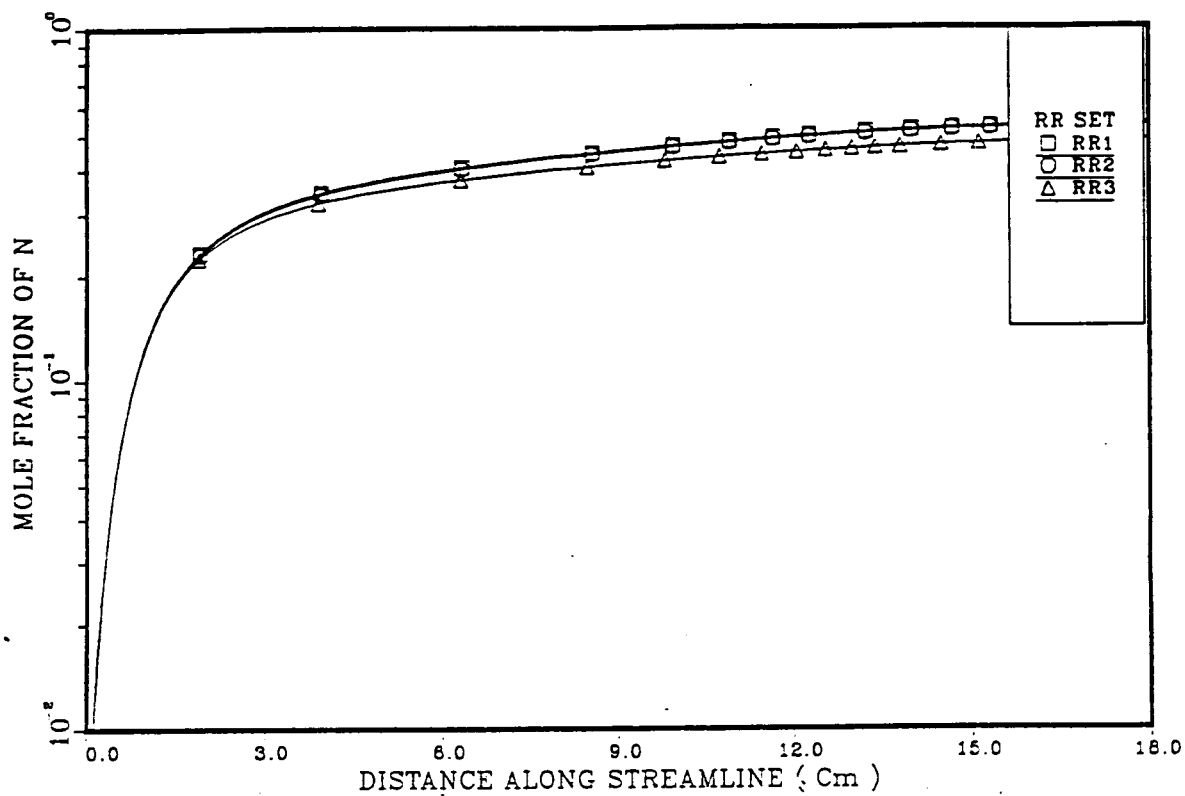


(a)

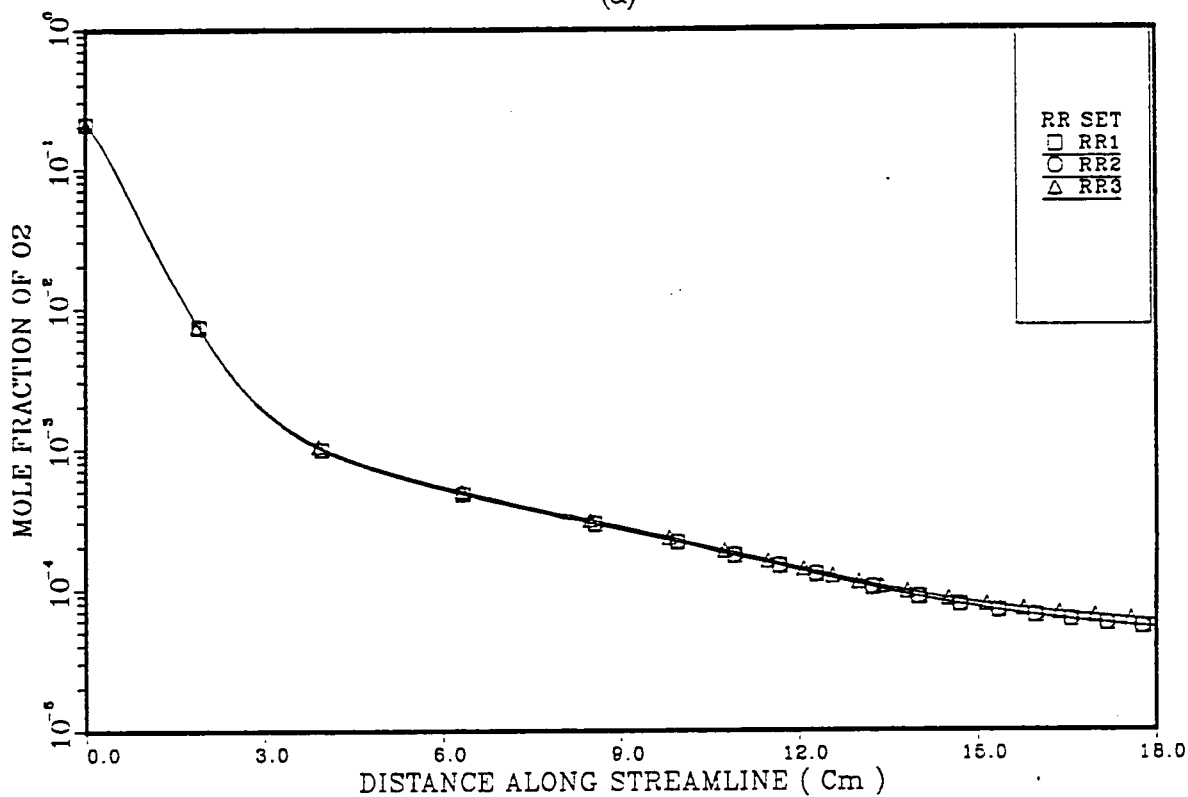


(b)

FIGURE 45. TVN2 AND N2 PROFILES AT V=7.7 Km/s, CVDV

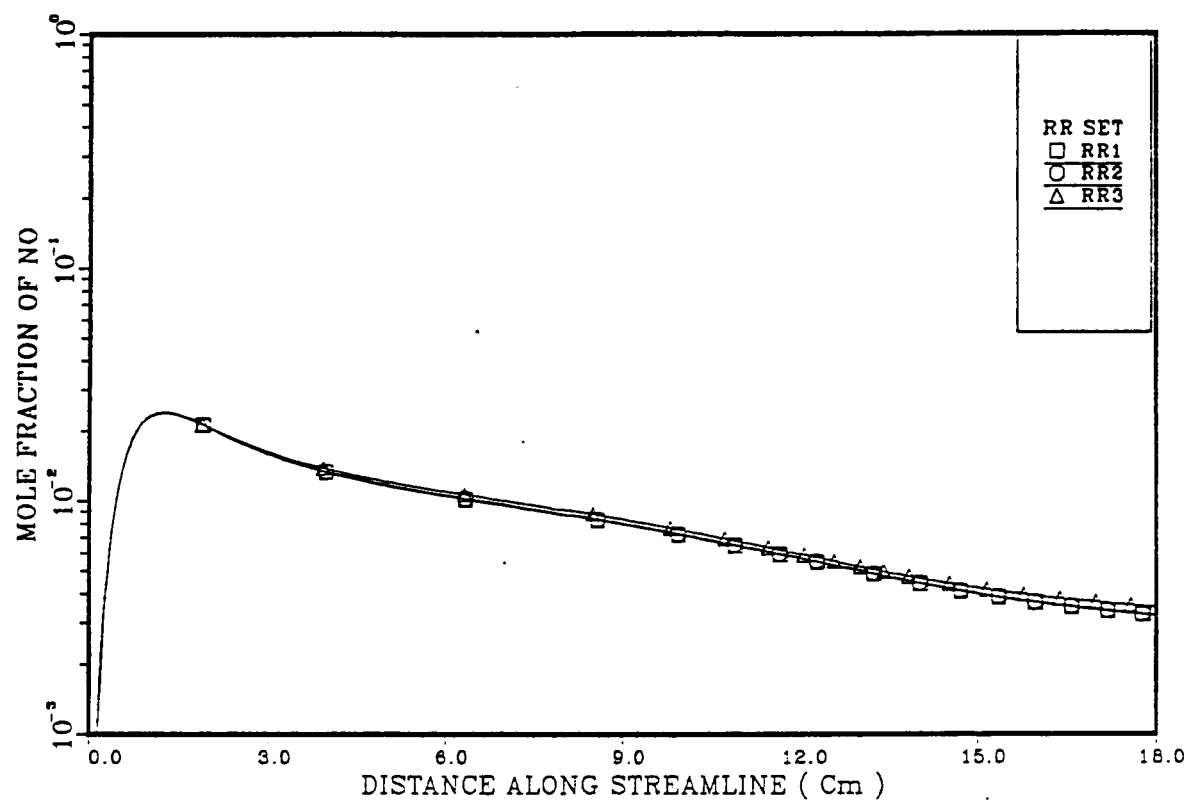


(a)

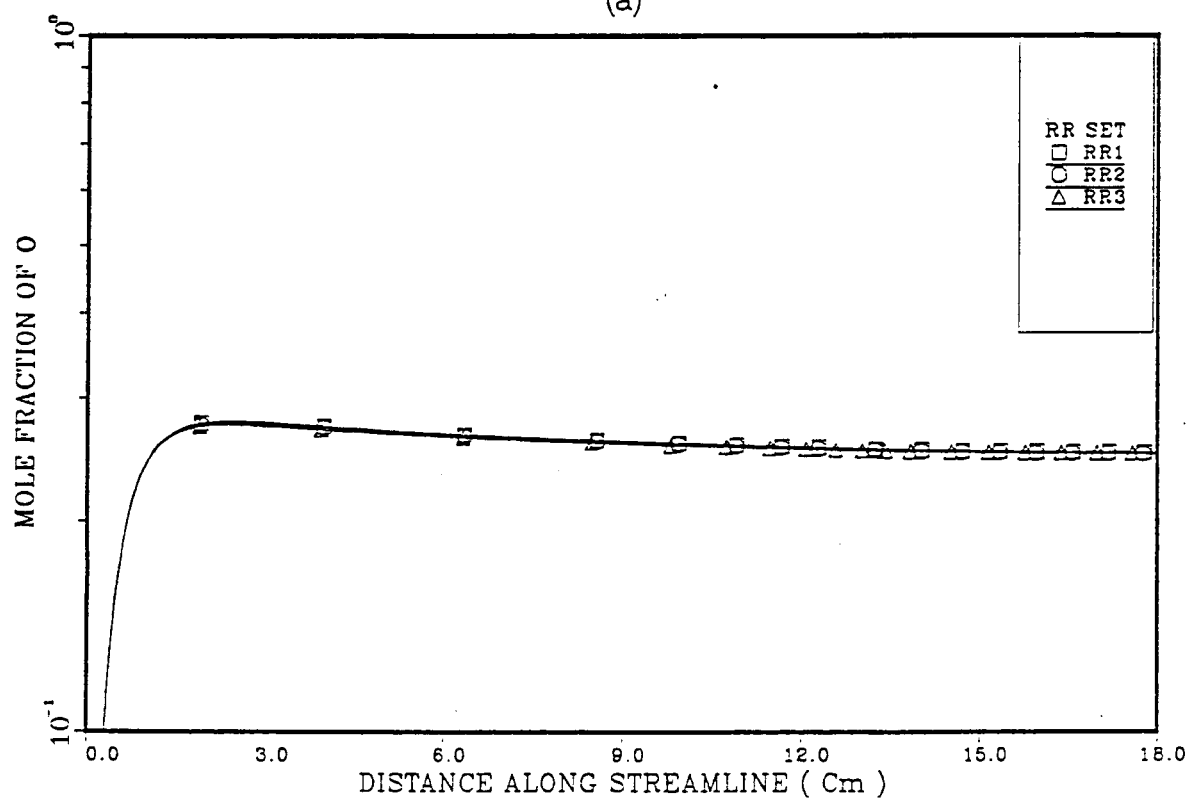


(b)

FIGURE 46. O2 AND N PROFILES AT V=7.7 Km/s, CVDV

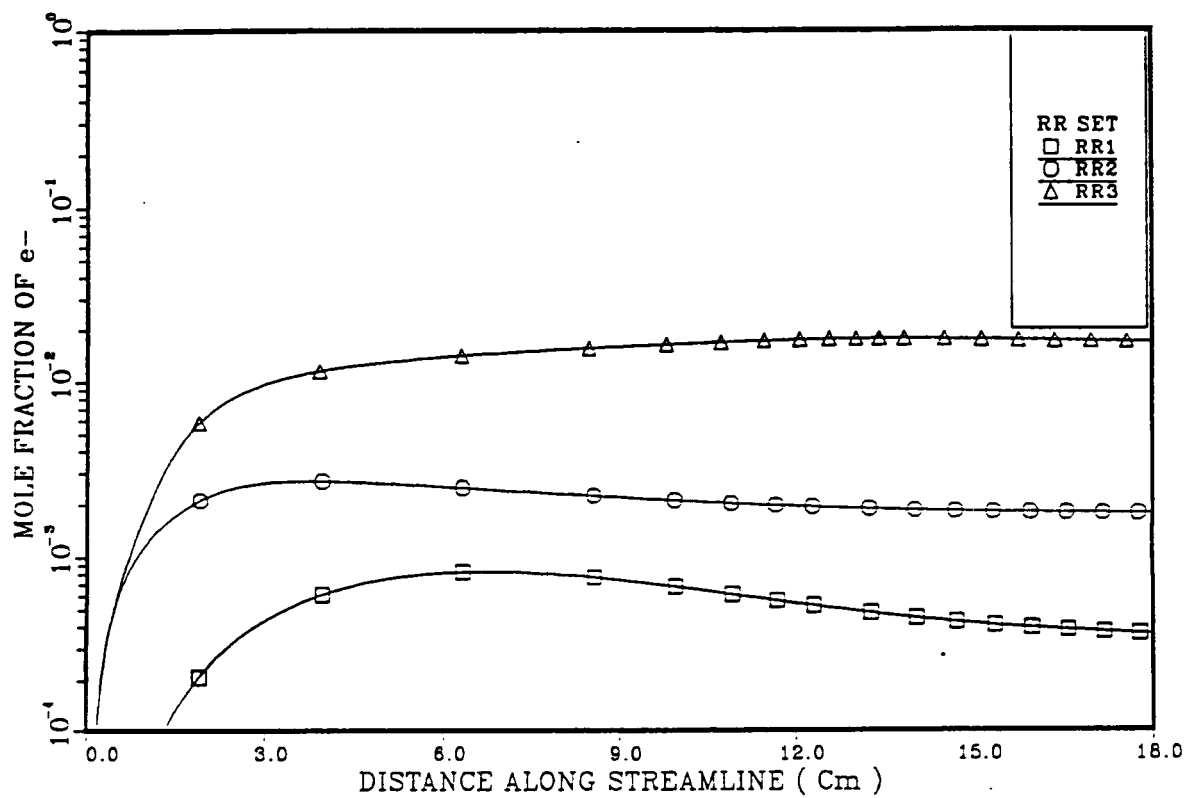


(a)

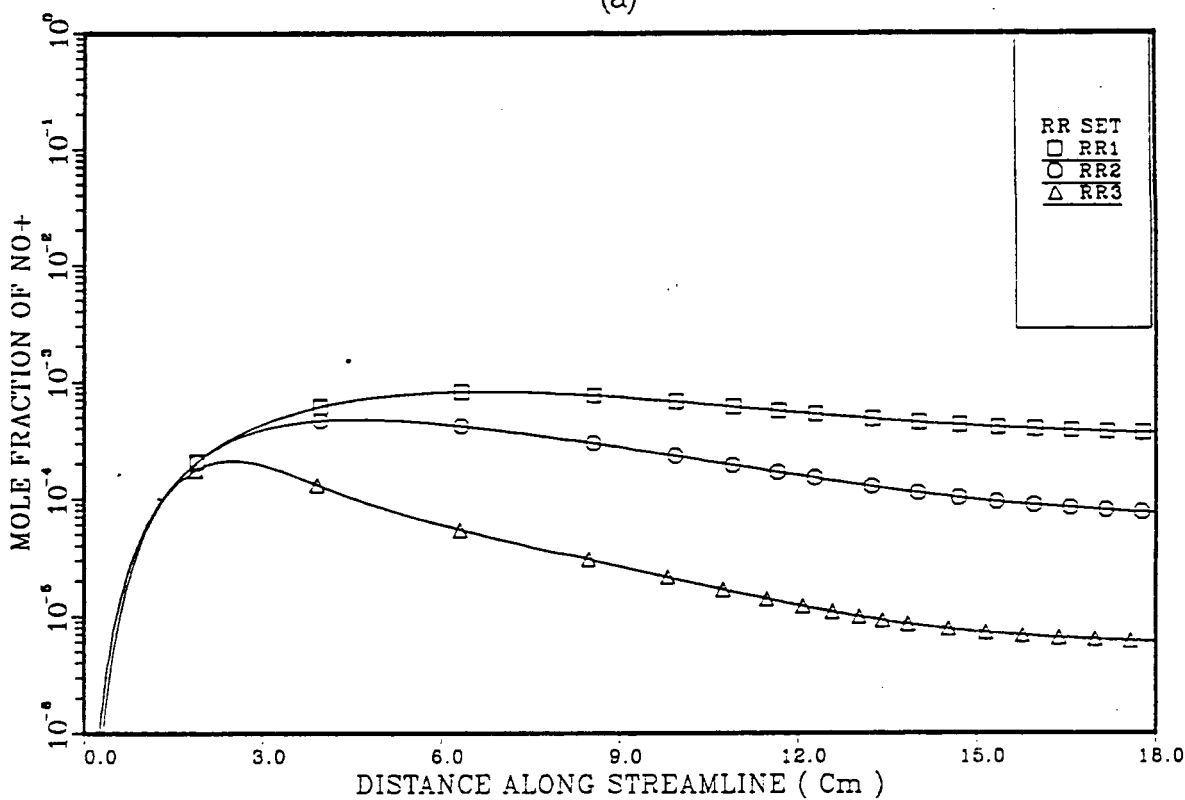


(b)

FIGURE 47. O AND NO PROFILES AT $V=7.7$ Km/s, CVDV

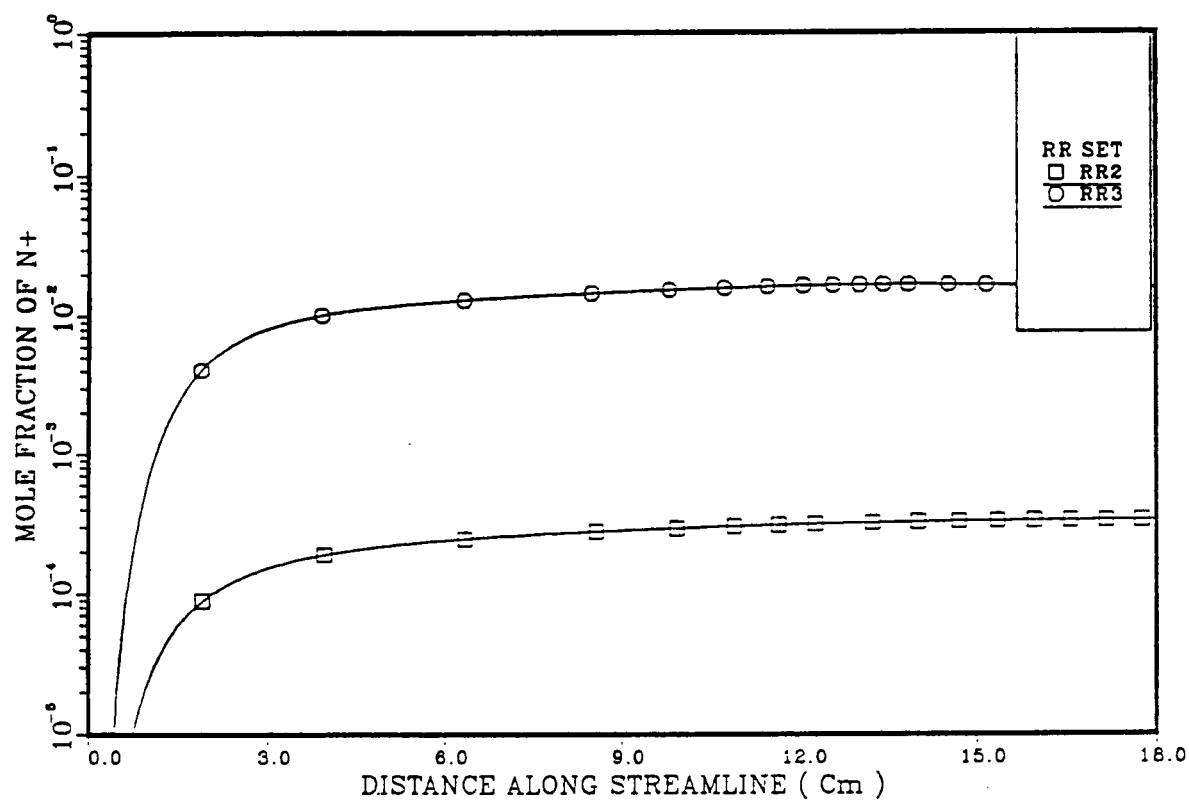


(a)

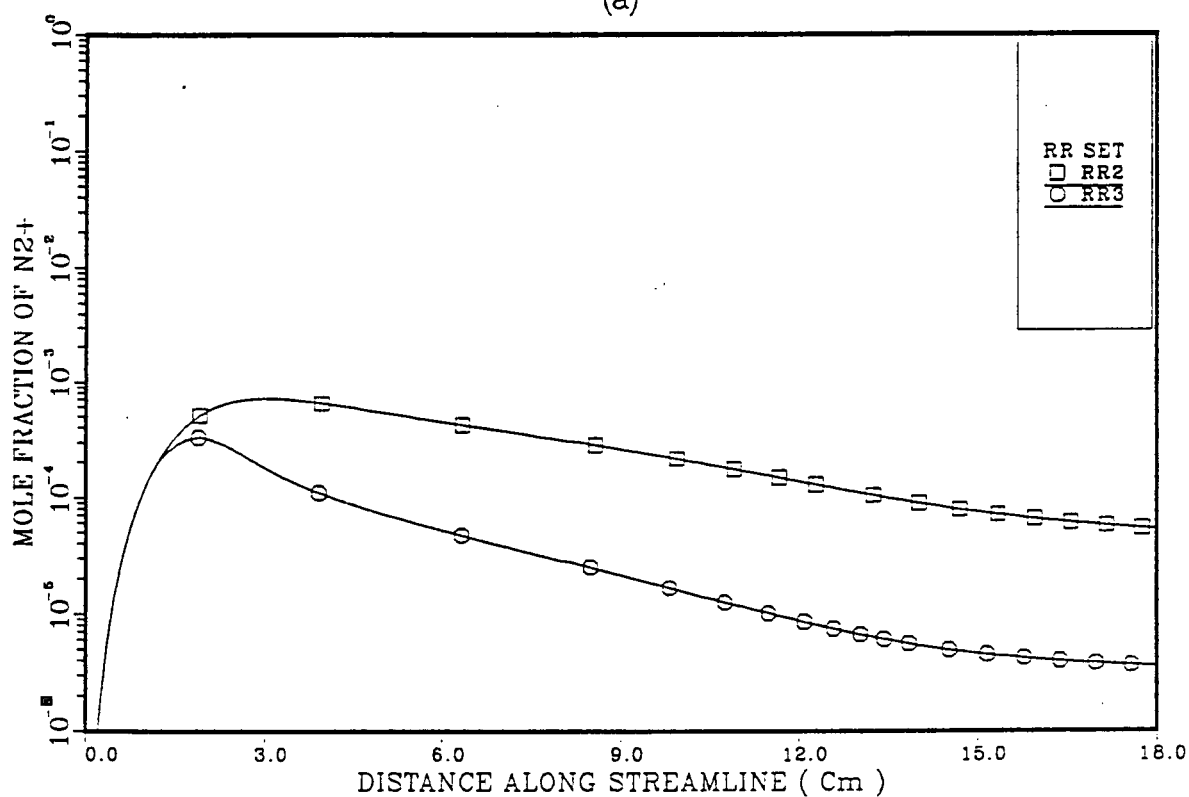


(b)

FIGURE 48. NO^+ AND e^- PROFILES AT $V=7.7$ Km/s, CVDV.

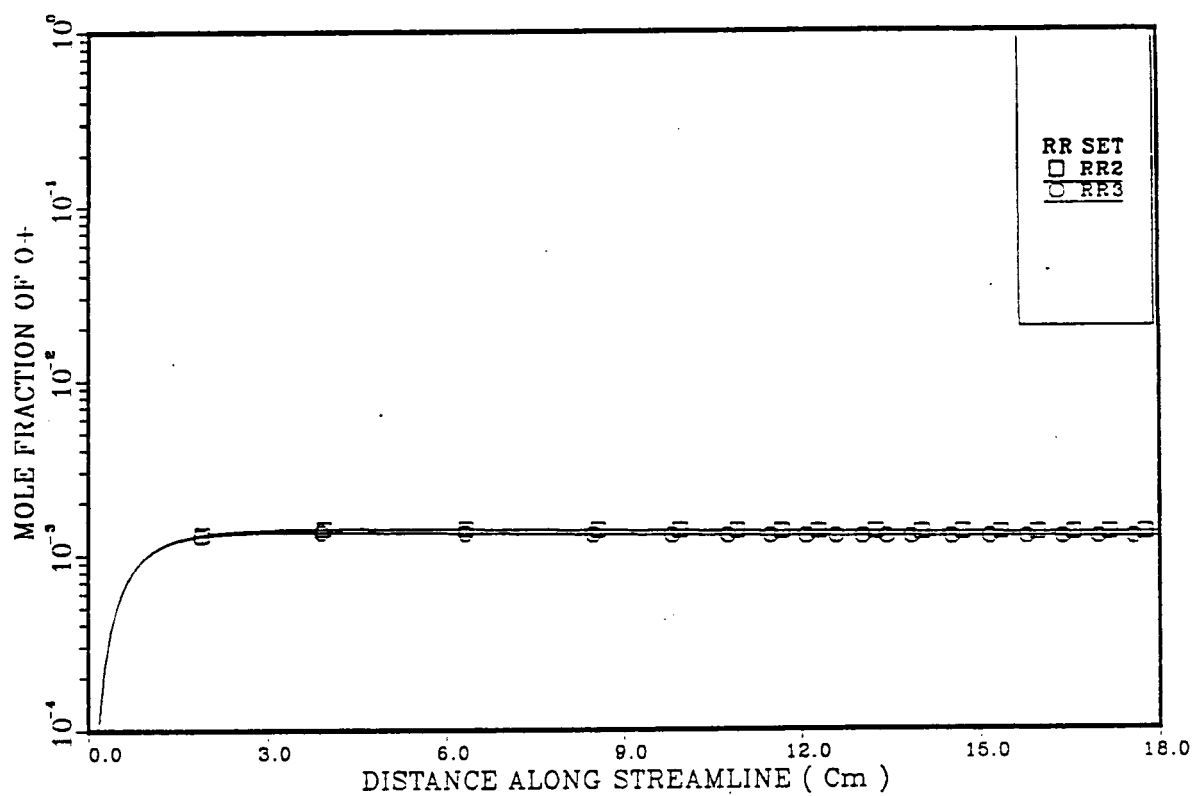


(a)



(b)

FIGURE 49. N_2^+ AND N^+ PROFILES AT $V=7.7$ Km/s, CVDV

FIGURE 50. O⁺ PROFILE AT V=7.7 Km/s, CVDV

CONCLUSIONS

The comparative study of reaction rates, species, and vibrational coupling models was a culmination of forty five computer runs. Only the profiles of physical significance were chosen and discussed for this thesis.

After much evaluation and discussion, the following conclusions were obtained from this comparative study. First, when the use of a vibrational coupling model is being considered, the coupling effect between the vibrational mode, dissociation, and recombination are extremely important. Therefore, the results obtained from the CVD model are considered to be incorrect since it fails to incorporate these effects. Secondly, as can be seen from the vibrational coupling and reaction rate studies, the substitution of different reaction rate coefficients for the electron impact rate had a substantial impact on all physical profiles. Additional reaction rate studies are needed to verify these coefficients along with other possible substitutions. Thirdly, the Park-Like model exhibited preferential dissociation, and a more accurate representation of electron temperature over a wider velocity range when compared to the CVD, CVDV, and CVDV-Preferential models. Finally, if ionization effects are not considered important, then the use of the smaller reaction rate set will predict reasonable answers with a substantial reduction in computational time.

REFERENCES

1. Dogra, V.K., Moss, J.N., and Simmonds, A.L., "Direct Simulation of Aerothermal Loads for an Aeroassist Flight Experiment Vehicle", AIAA Paper 87-1546.
2. Park, C., "Radiation Enhancement by Non-Equilibrium in Earth's Atmosphere", AIAA Paper 83-0410.
3. Maslen, S.H., "Inviscid Hypersonic Flow Past Smooth Symmetric Bodies", AIAA Journal, Vol. 2, No. 6, June 1964.
4. Grose, W.L., "A Thin-Shock Layer Solution for Non-Equilibrium Inviscid Hypersonic Flows in Earth, Martian, and Venusian Atmospheres", NASA TN D-6529, December 1971.
5. Hammerling, P., Teare, J.D., and Kivel, B., "Theory of Radiation from Luminous Shock Waves in Nitrogen", Physics of Fluids, Vol. 2, No. 4, July-August 1959.
6. Treanor, C.E., and Marrone, P.V., "Effect of Dissociation on the Rate of Vibrational Relaxation", Physics of Fluids, Vol. 5, No. 9, September 1962.
7. Garr, L.J., and Marrone, P.V., "Inviscid Non-Equilibrium Flow Behind Bow and Normal Shock Waves", Parts I, II, Cornell Aeronautical Laboratory, CAL Rpt. QM-1626-A-12, May 1963.
8. Marrone, P.V., and Treanor, C.E., "Chemical Relaxation with Preferential Dissociation from Excited Vibrational Levels", Physics of Fluids, Vol. 6, No. 9, September 1963.
9. Herzburg, G., Spectra of Diatomic Molecules, D. Van Nostrand Co., Inc., New York, 1950.
10. Park, C., "Assessment of Two-Temperature Kinetic Model for Ionizing Air", AIAA Paper 87-1574, June 1987.

11. Landau, L., and Teller, E., "Physik. Z. Sowjetunion", 10, 34, 1936 in Grose, W.L., NASA TN D-6529, December 1971.
12. Millikan, R.C., and White, D.R., "Systematics of Vibrational Relaxation", Journal of Chemical Physics, Vol. 39, No. 12, December 1963.
13. Blackman, V.L., "Vibrational Relaxation in Oxygen and Nitrogen", Journal of Fluid Mechanics, Vol. 1, 1956.
14. Hanson, R.K., and Bagonoff, D., "Shock-Tube Study of Vibrational Relaxation in Nitrogen Using Pressure Measurements", Journal of Chemical Physics, Vol. 53, 1970.
15. Greendyke, R.B., "A Parametric Study of Shock Jump Chemistry, Electron Temperature, and Radiative Heat Transfer Models in Hypersonic Flows", M.S. Thesis, Texas A&M University, August 1988.
16. Treanor, C.E., "A Method for the Numerical Integration of Coupled First Order Differential Equations with Greatly Different Time Constants", Cornell Aeronautical Laboratory, CAL Rpt. AG-1729-A-4, September 1963.
17. Vincenti, W.G., and Kruger, C.H., Introduction to Physical Gas Dynamics, Robert E. Krieger Publishing Co., Malabar, Florida, 1965.
18. Carlson, L.A., "Radiative-Gasdynamic Coupling and Nonequilibrium Effects behind Reflected Shock Waves", AIAA Journal, Vol. 9, No. 5, May 1971.
19. Carnicom, M.L., "Reaction Rates for High Temperature Air", Sandia Rpt. SC-R-66-885, May 1966.
20. Kang, S.W., Dunn, M.G., and Jones, W.L., "Theoretical and Measured Electron-Density Distributions for the Ram Vehicle at High Altitudes", AIAA Paper 72-689.

APPENDIX

The coupling factor is derived using the assumption that the vibrational energy levels are modeled as a system of harmonic oscillators which are cutoff just prior to dissociation. The partition function which represents this concept is

$$Q(T) = \frac{1 - e^{(-\theta_d/T)}}{1 - e^{(-\theta_v/T)}} \quad (76)$$

The coupling factor is defined as a ratio of the actual to the local vibrational equilibrium forward reaction rates. The coupling factor can then be redefined in the form of partition functions as

$$\phi = \frac{k_f(T, T_v)}{k_f(T_v)} = \frac{Q(T) Q(T_m)}{N Q(T_v)} \quad (77)$$

Substituting equation (76) into (77) yields

$$\phi = \frac{1}{N} \left[\frac{1 - e^{(-\theta_d/T)}}{1 - e^{(-\theta_v/T)}} \right] \left[\frac{1 - e^{(-\theta_d/T_m)}}{1 - e^{(-\theta_v/T_m)}} \right] \cdot \left[\frac{1 - e^{(-\theta_d/T_v)}}{1 - e^{(-\theta_v/T_v)}} \right] \quad (78)$$

The characteristic dissociation temperature (θ_d) is assumed to be negligible for the $(-1/T)$ and $(-1/T_v)$ terms during dissociation. Therefore equation (78) is reduced to

$$\phi = \frac{1}{N} \left[\frac{1 - e^{(-\theta_v/T_v)}}{1 - e^{(-\theta_v/T)}} \right] \left[\frac{1 - e^{(-\theta_d/T_m)}}{1 - e^{(-\theta_v/T_m)}} \right] \quad (79)$$

The final form of this equation is obtained by multiplying the first factor by $(e^{[\theta_v/T]}/e^{[\theta_v/T]})$ and the second factor by $(e^{[\theta_v/T_m]}/e^{[\theta_v/T_m]})$ which yields

$$\phi = \frac{1}{N} \left[\frac{1 - e^{(-\theta_d/T_m)}}{e^{(\theta_v/T_m)} - 1} \right] \left[\frac{e^{(\theta_v/T_v)} - 1}{e^{(\theta_v/T)} - 1} \right] \quad (80)$$

VITA

Glenn James Bobskill was born in [REDACTED] [REDACTED]
[REDACTED] [REDACTED] [REDACTED] After obtaining a GED in 1976, he
enlisted in the United States Navy and served as a Naval
Aircraft Ordnanceman at NAS Moffett Field between April of
1977 and April of 1981. After attending Odessa Jr. College
for a year, he was accepted to Texas A&M University and
obtained a B.S. degree in Aerospace Engineering in May of
1986. He was then accepted to the Graduate College at Texas
A&M in September of 1986 in the Department of Aerospace
Engineering. He will receive his M.S. degree in August of
1988 and expects to continue his work as a Research Engineer
within the aerospace community.

APPENDIX 3

Compendium of Reaction Rate, Species, and Vibration-Dissociation Coupling
Results

Compiled by Glenn J. Bobskill

NOTES

I. All streamline results are for the streamline crossing the shock front nine centimeters above the axis of a blunt body having a nose radius of approximately 230 cm..

II. Figures 5 - 34 are for reaction rate system RR1 described in AIAA Paper No. 88-2673 in Appendix 1.

III. Figures 35 - 70 are for reaction rate system RR2 described in AIAA Paper No. 88-2673 in Appendix 1.

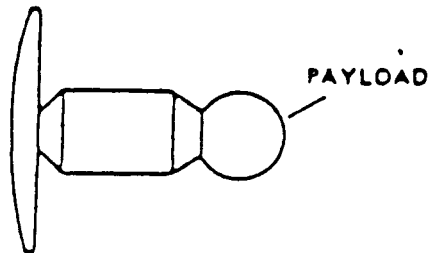
IV. Figures 71 - 106 are for reaction system RR3 described in AIAA Paper No. 88-2673 in Appendix 1.

V. Figures 107 - 125 are comparisons of the three reaction systems at three different velocities, all with CVDV coupling.

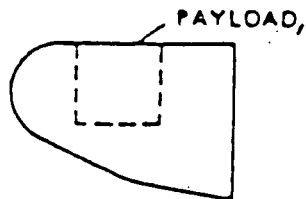
VI. Figures 126 - 142 are comparisons of the three reactions systems at three different velocities, all with MCVDV coupling.

VII. On many of the figures the legend uses the notation "Park-L". This is identical to the notation "MCVDV" used in AIAA Paper 88-2673.

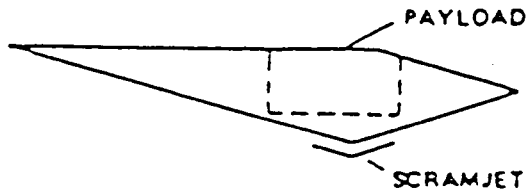
ORIGINAL PAGE IS
OF POOR QUALITY



1 (a) LOW-LIFT (AERO-BRAKING) BALLISTIC



1 (b) MEDIUM-LIFT BALLISTIC



1 (c) HIGH-LIFT AIRBREATHING

FIGURES 1(a),1(b),1(c).GENERIC AOTV CONFIGURATIONS
(Adapted from Reference 14)

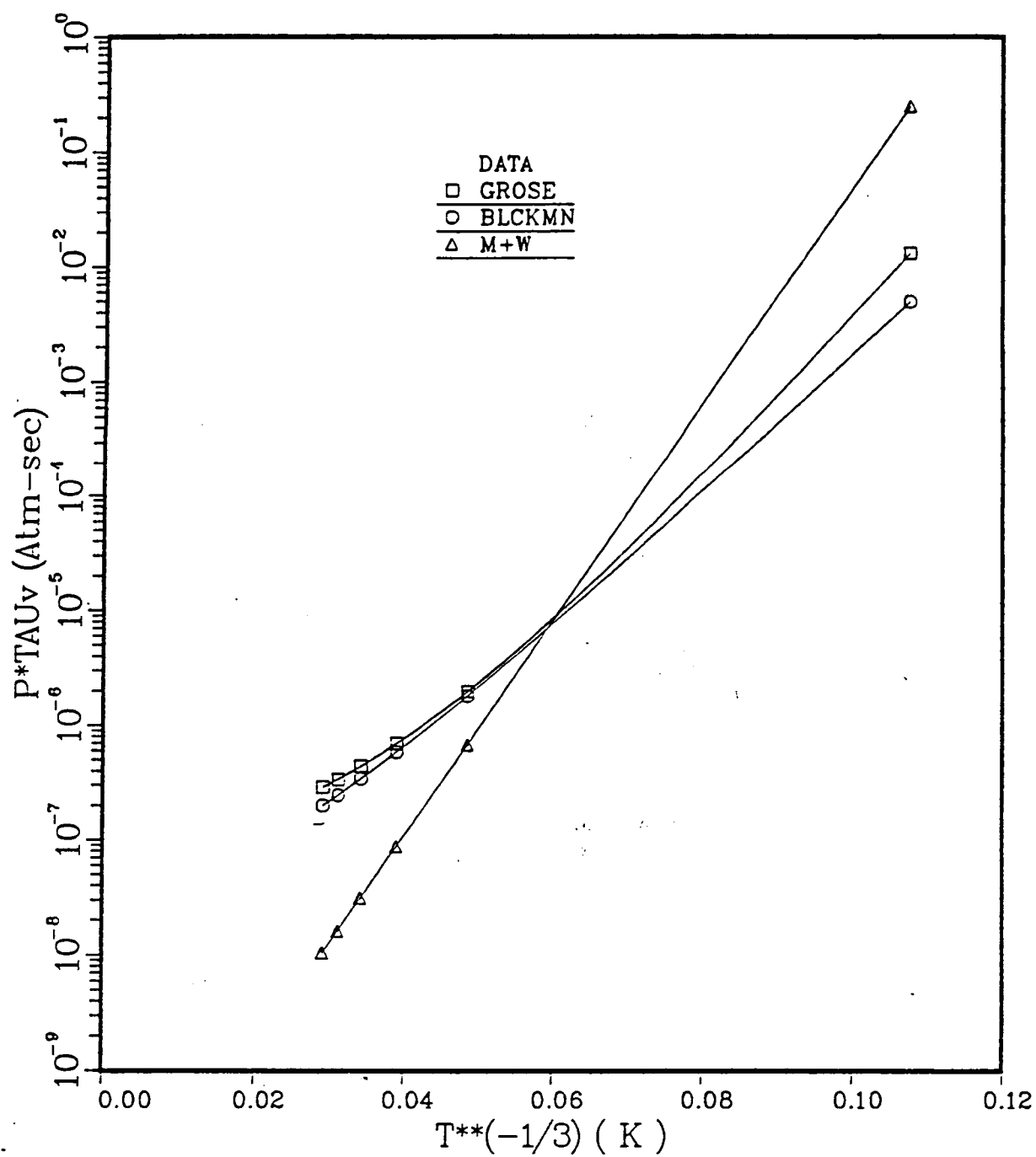


FIGURE 2. VIBRATIONAL RELAXATION DATA FOR NITROGEN

ORIGINAL PAGE IS
OF POOR QUALITY

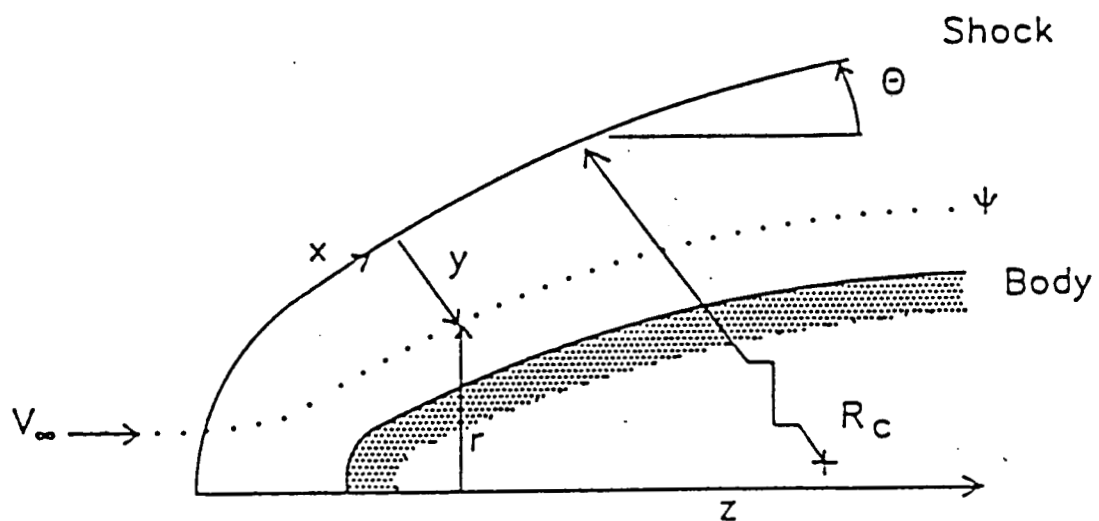


FIGURE 3. SHOCK-ORIENTED COORDINATE SYSTEM

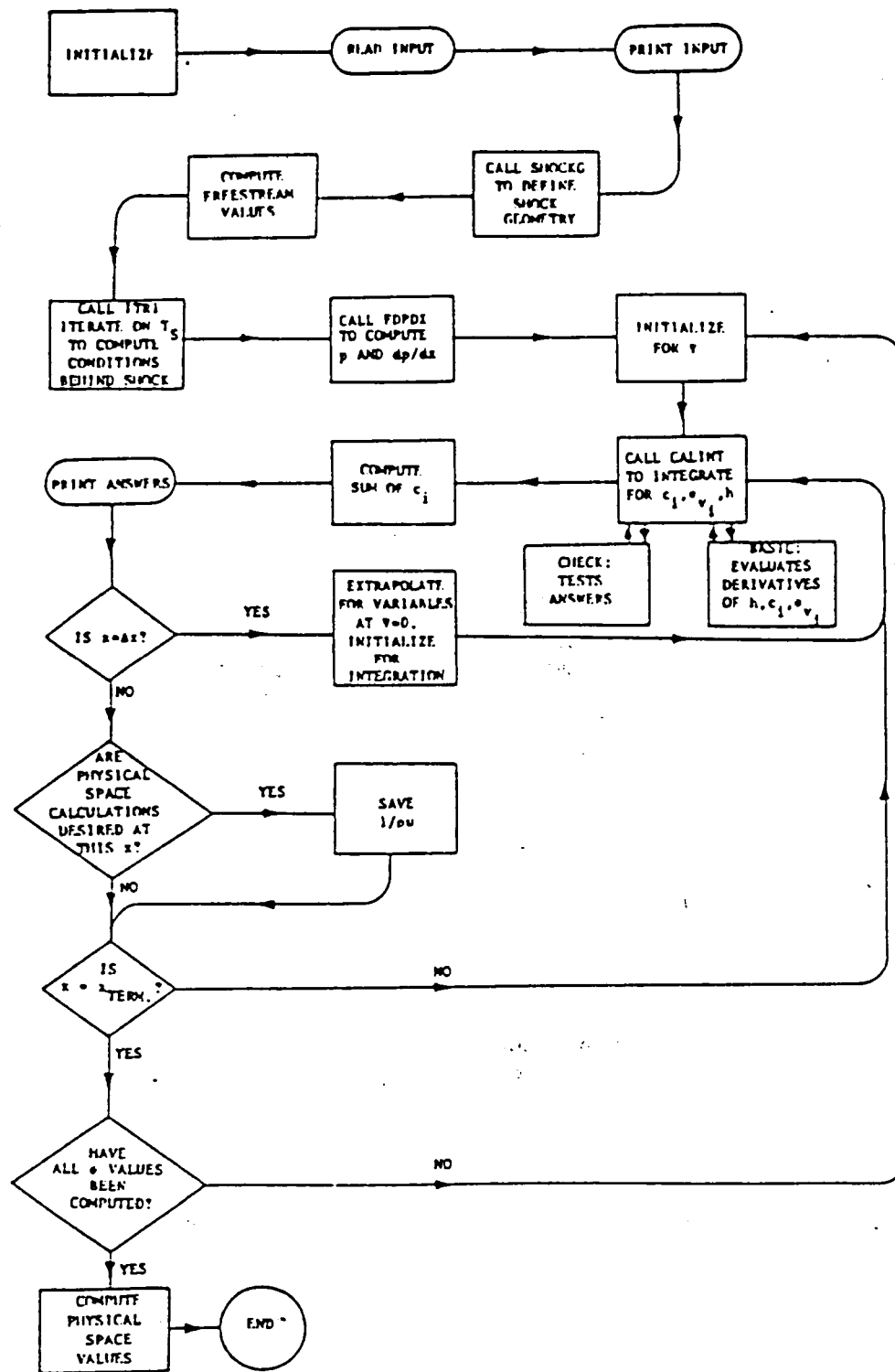
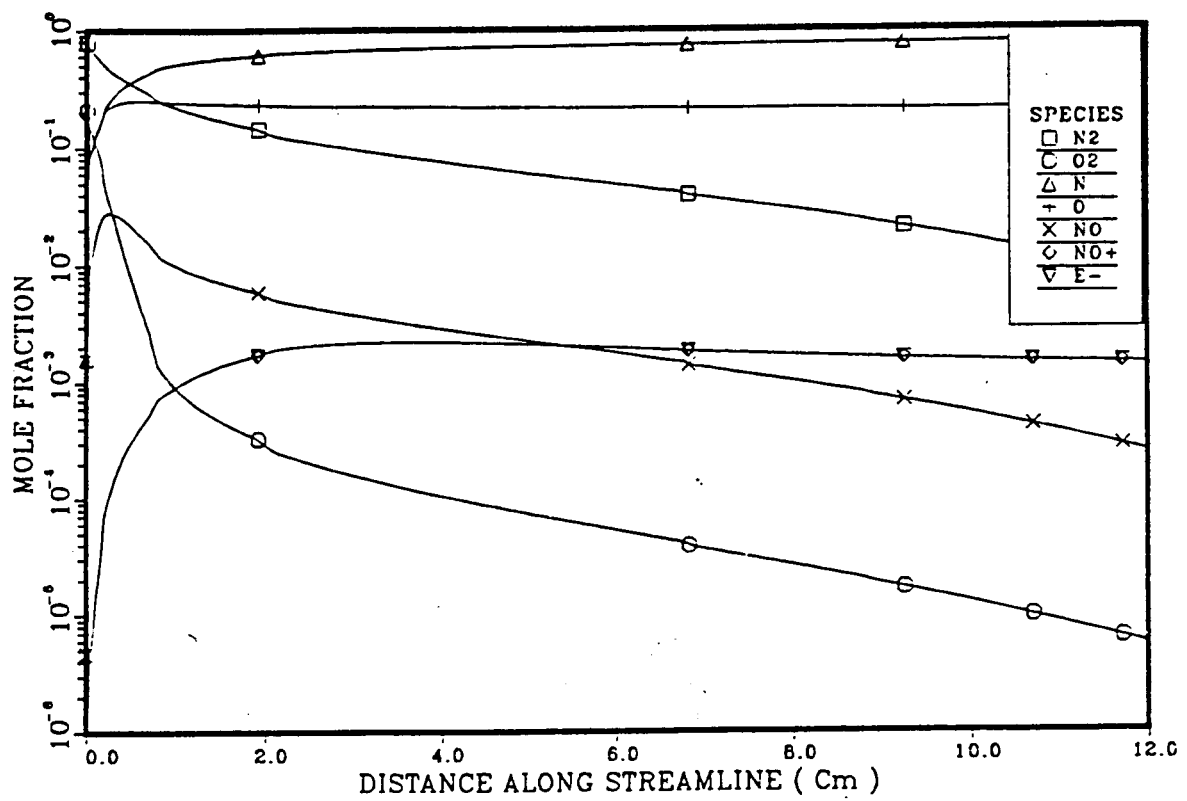
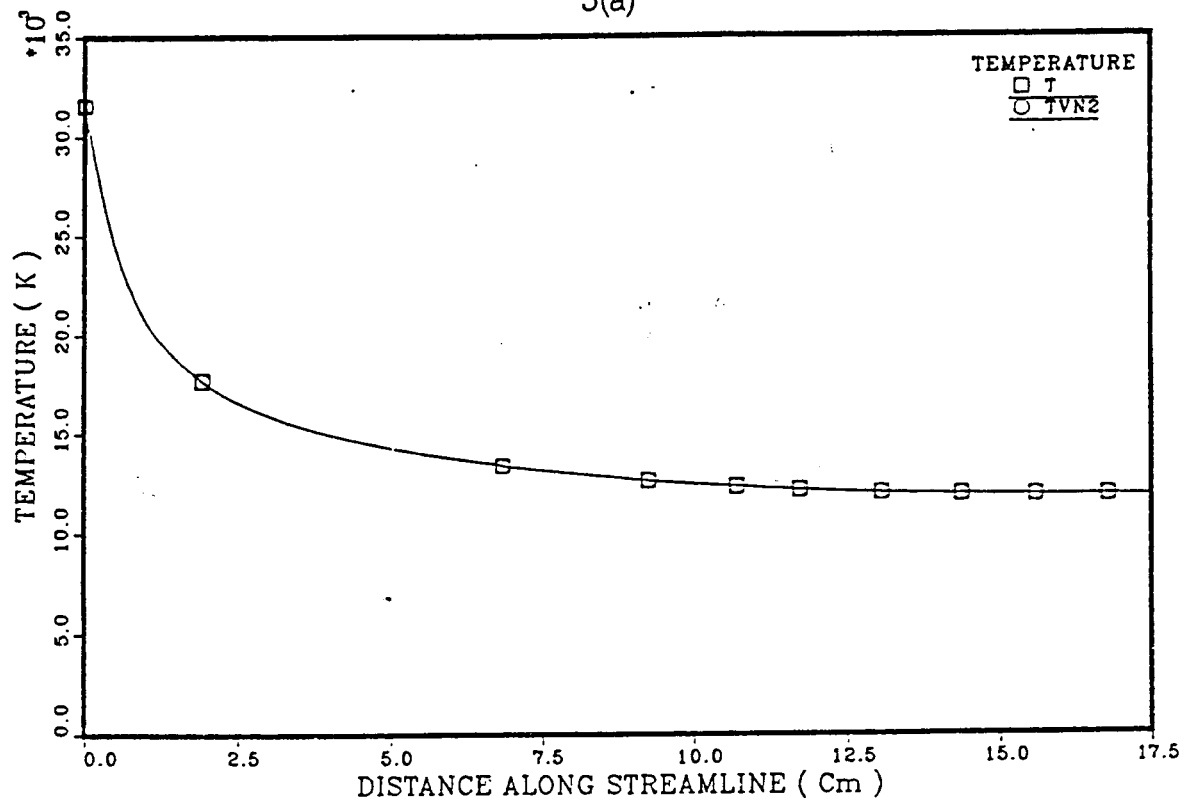


FIGURE 4.FLOW CHART OF SOLUTION ALGORITHM
(Adapted from Reference 2)

ORIGINAL PAGE IS
OF POOR QUALITY

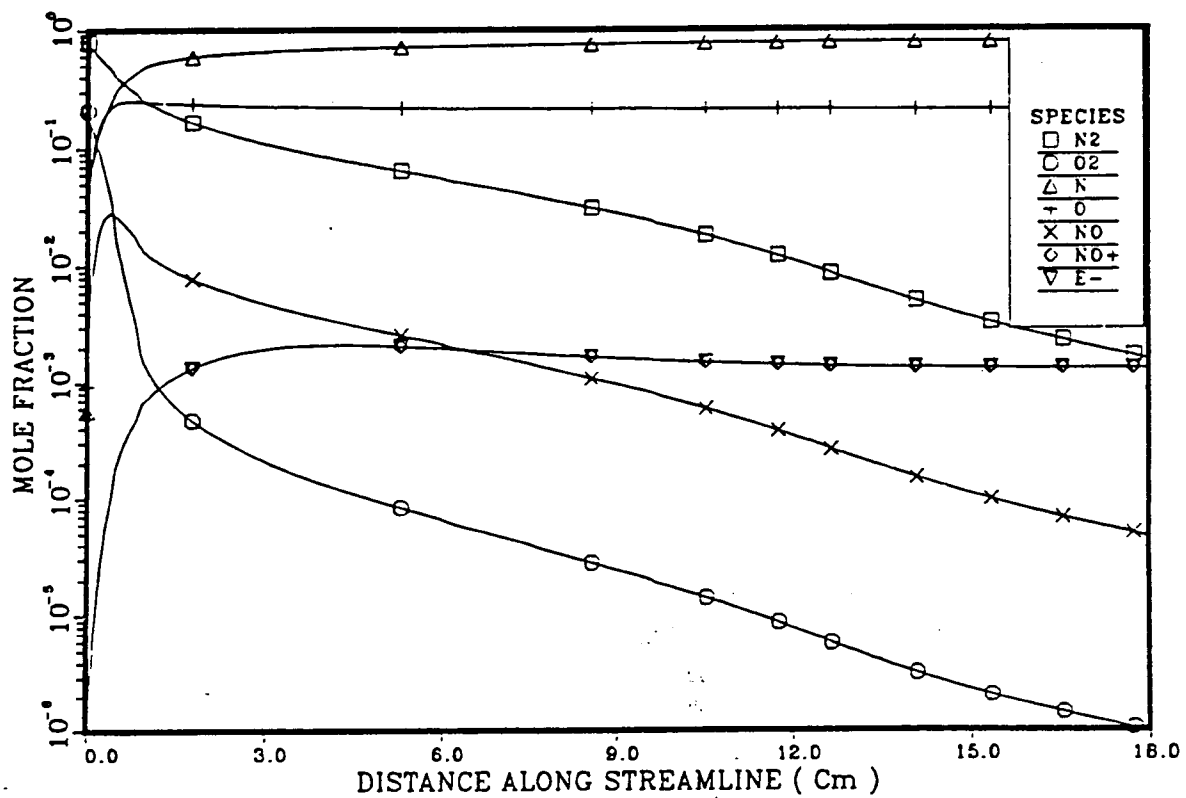


5(a)

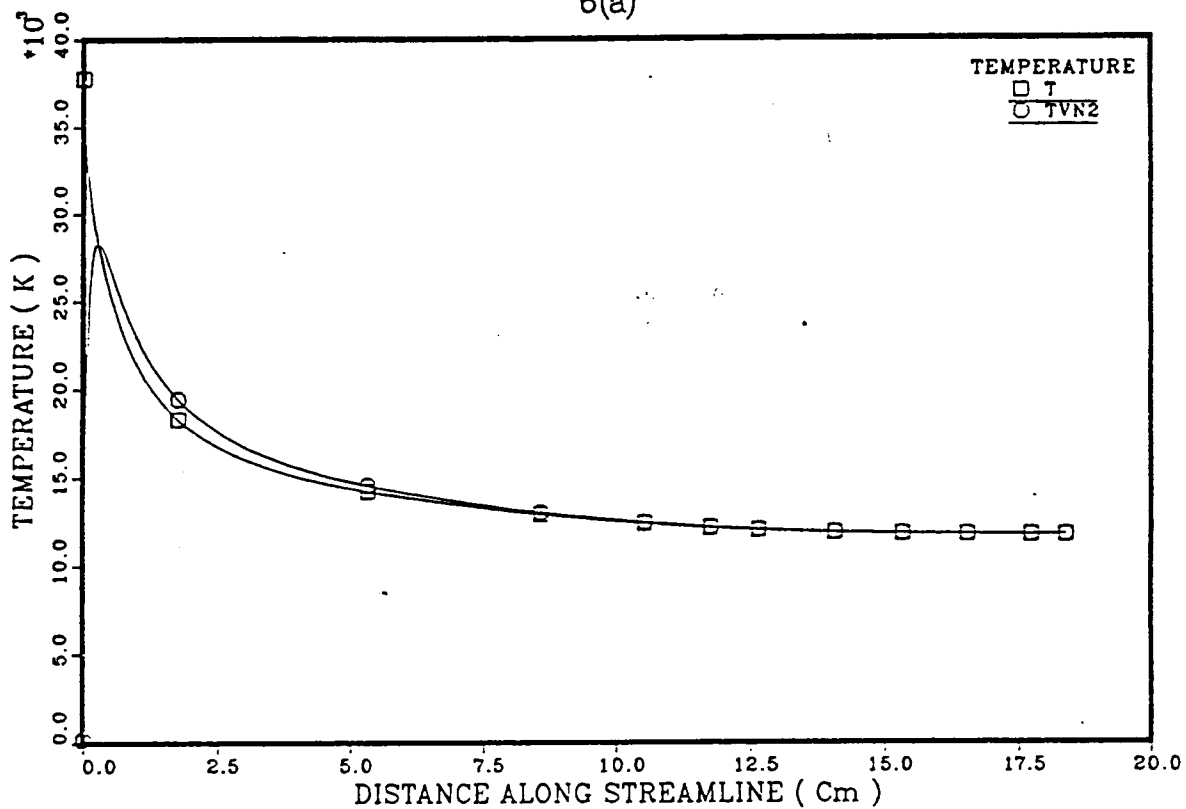


5(b)

FIGURES 5(a),5(b).VIB-EQ MODEL AT $V=10$ Km/s, RR1



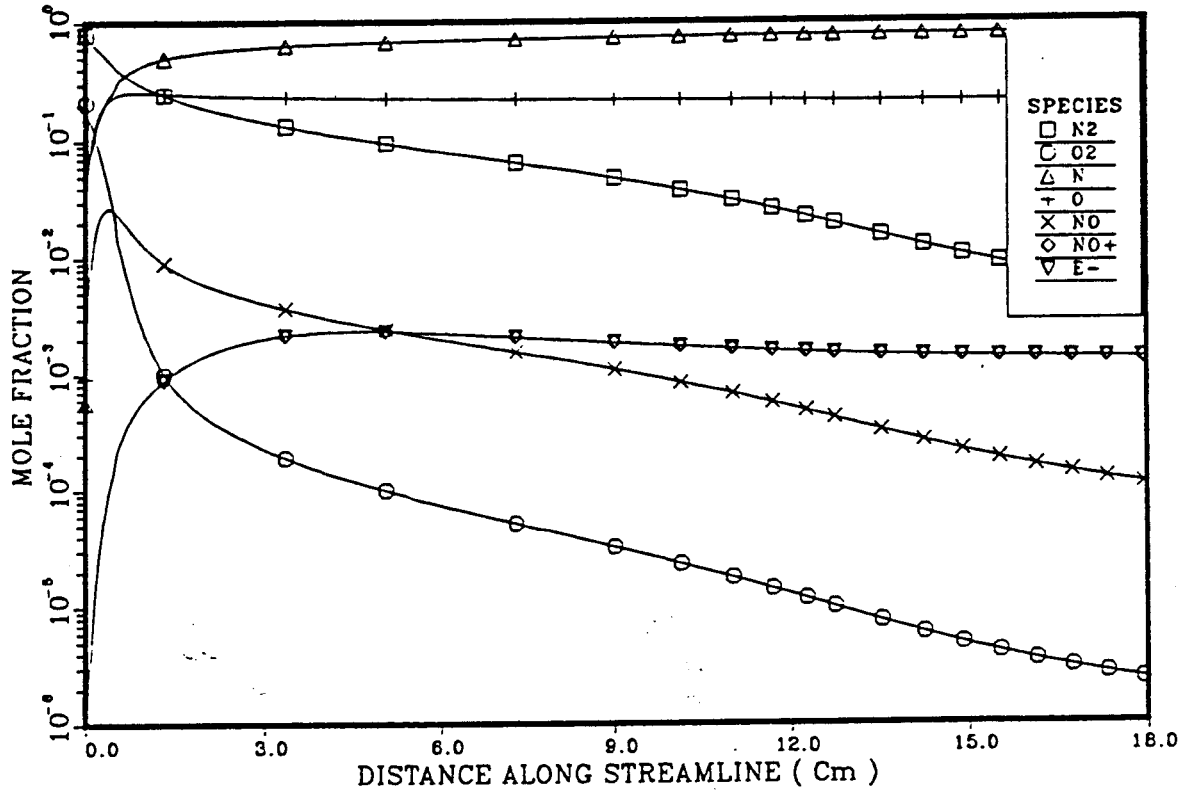
6(a)



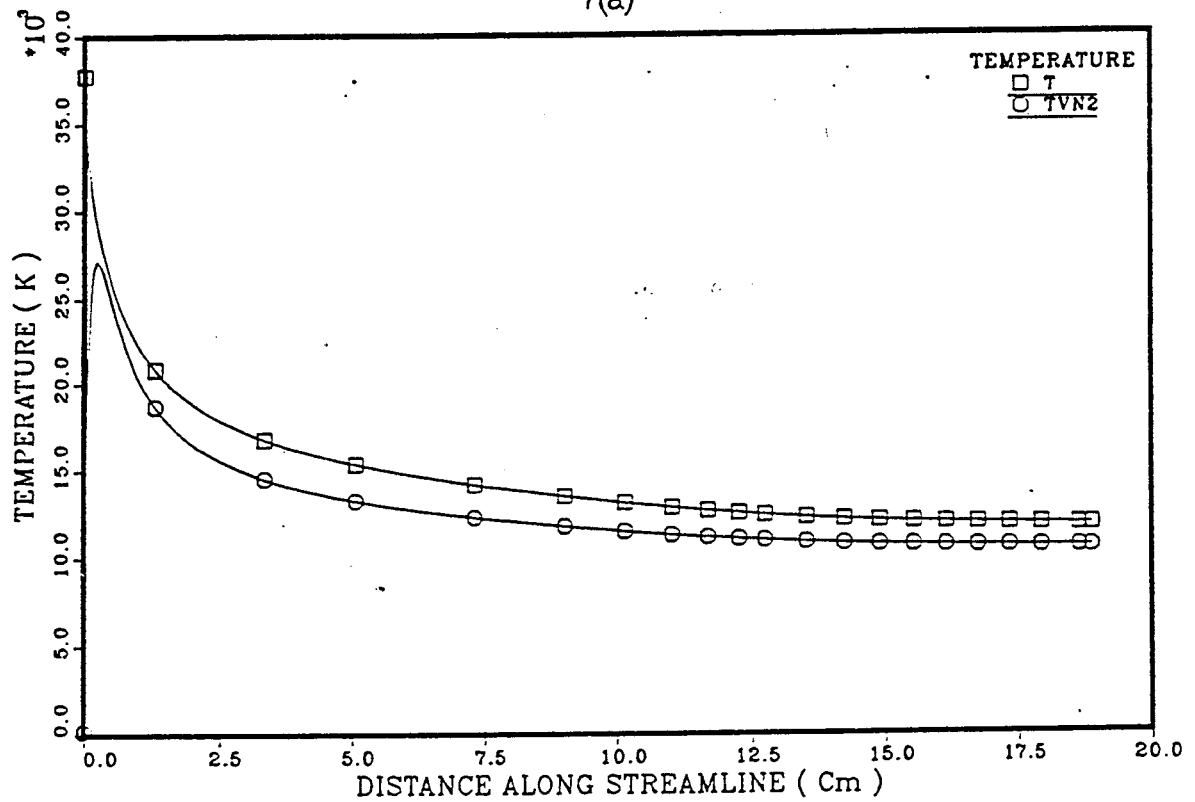
6(b)

FIGURES 6(a),6(b).CVD MODEL AT V=10 Km/s, RR1

ORIGINAL PAGE IS
OF POOR QUALITY

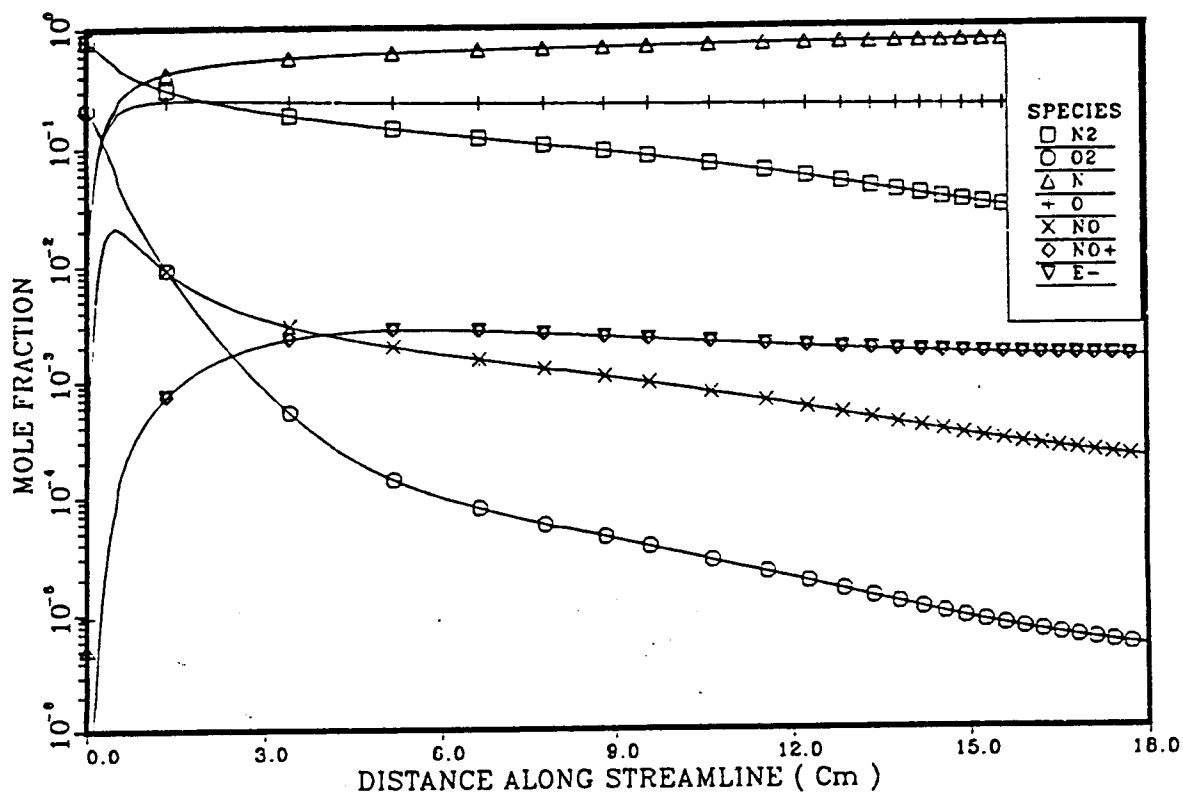


7(a)

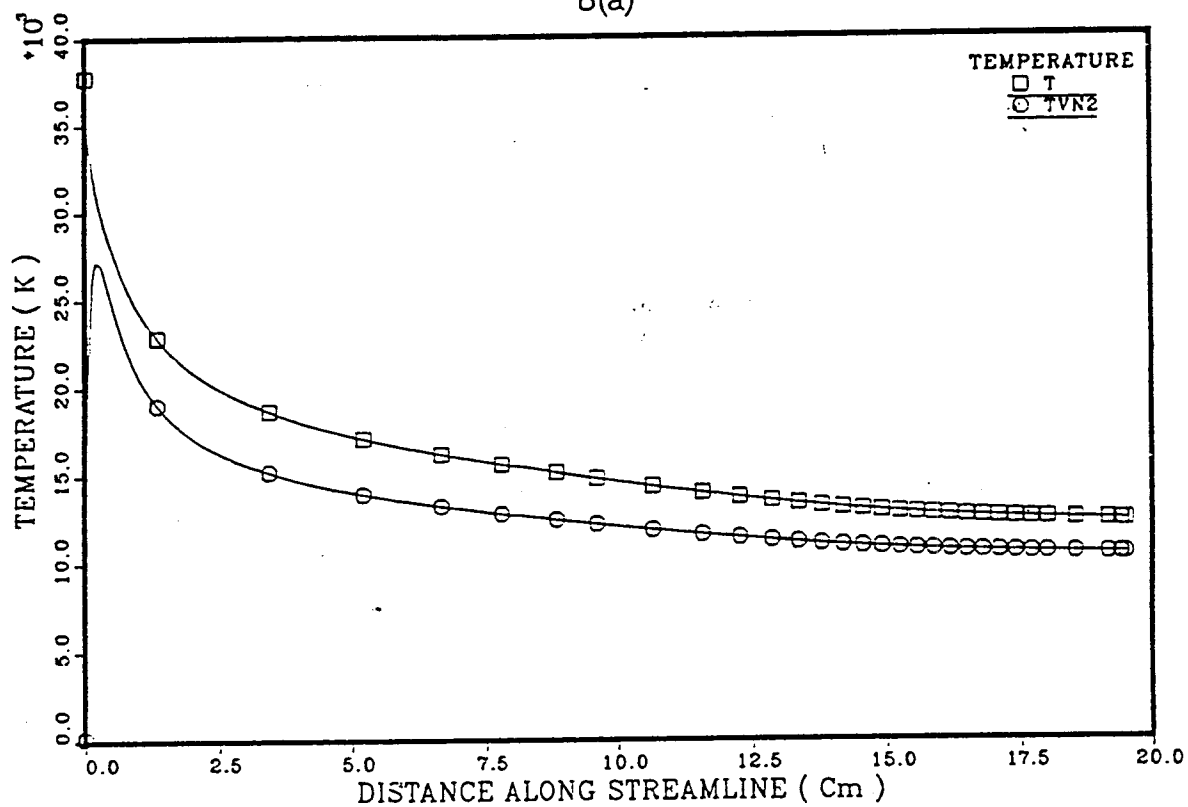


7(b)

FIGURES 7(a),7(b).CVDV MODEL AT V=10 Km/s, RR1

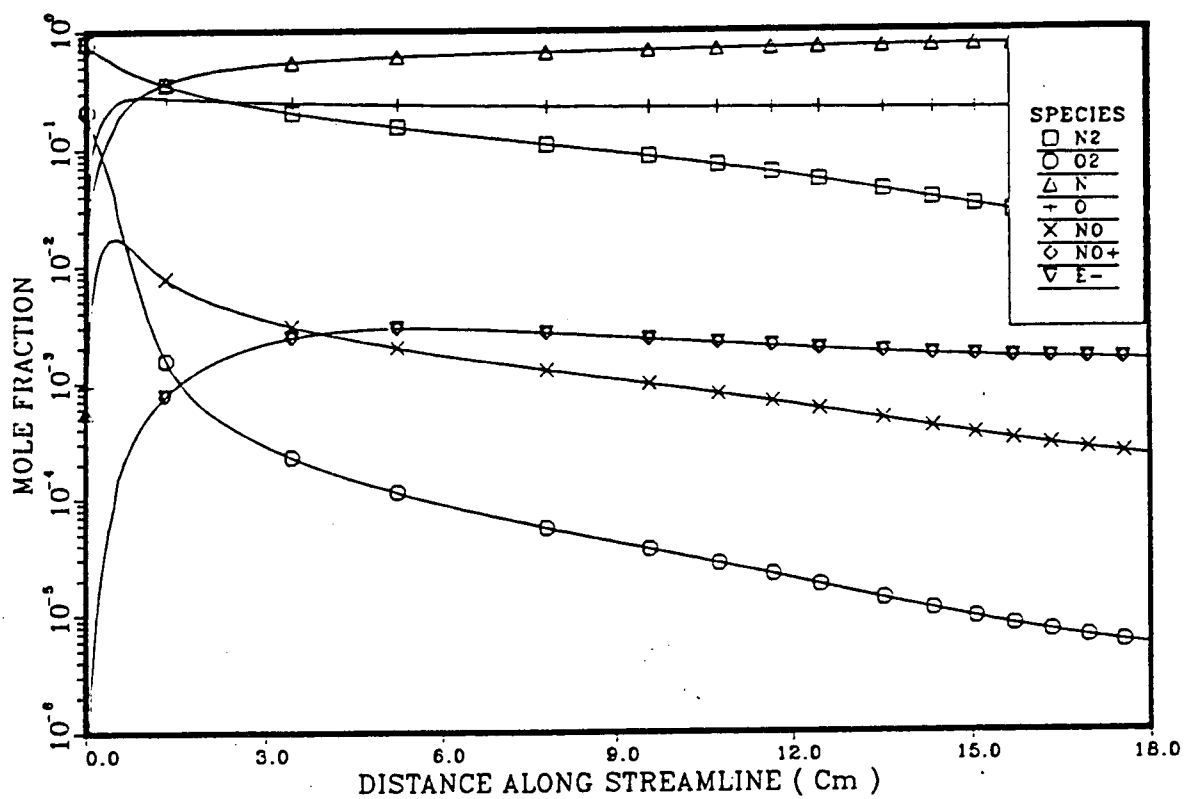


8(a)

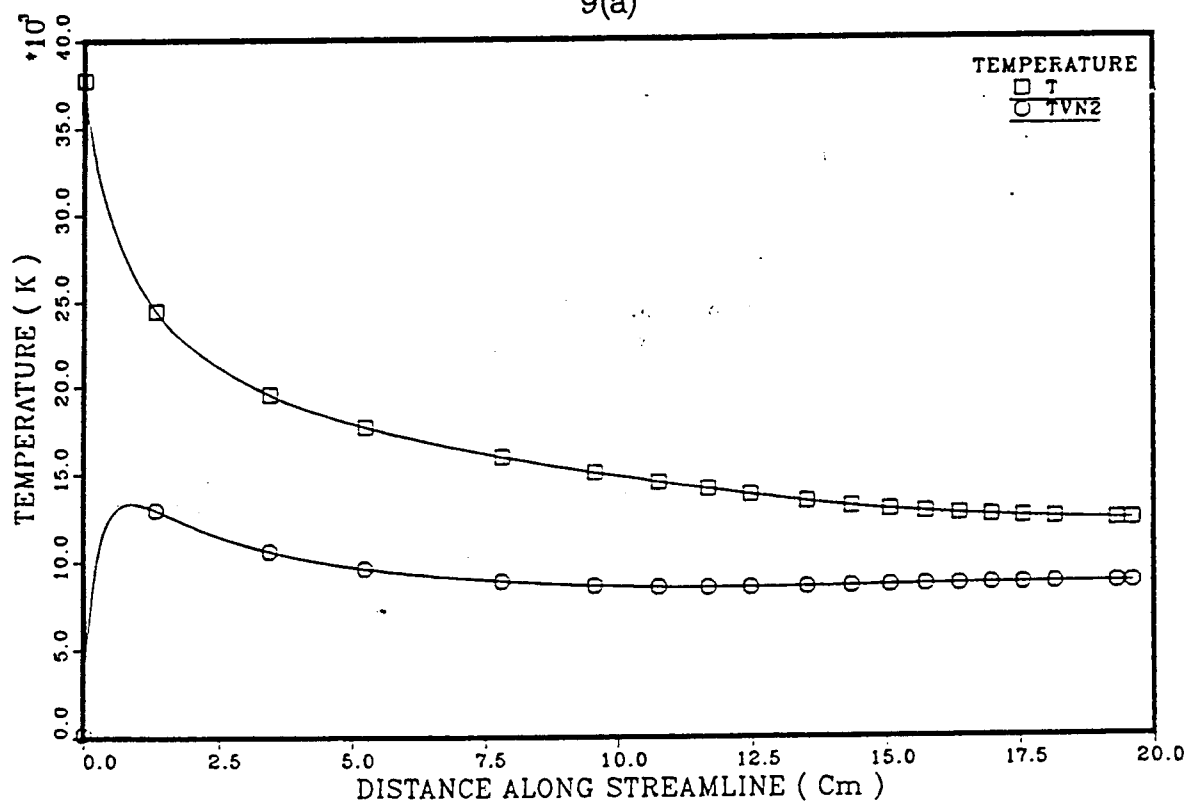


8(b)

FIGURES 8(a),8(b).CVDV-P MODEL AT V=10 Km/s, RR1

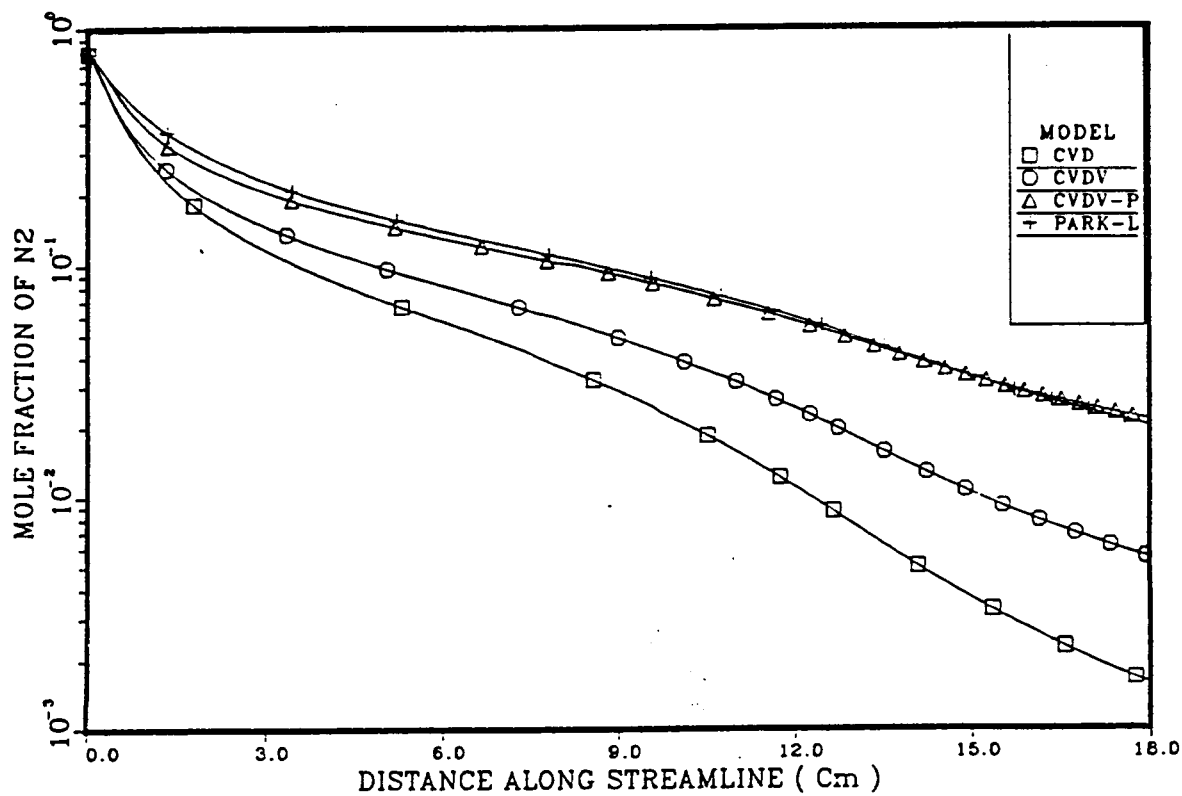


9(a)

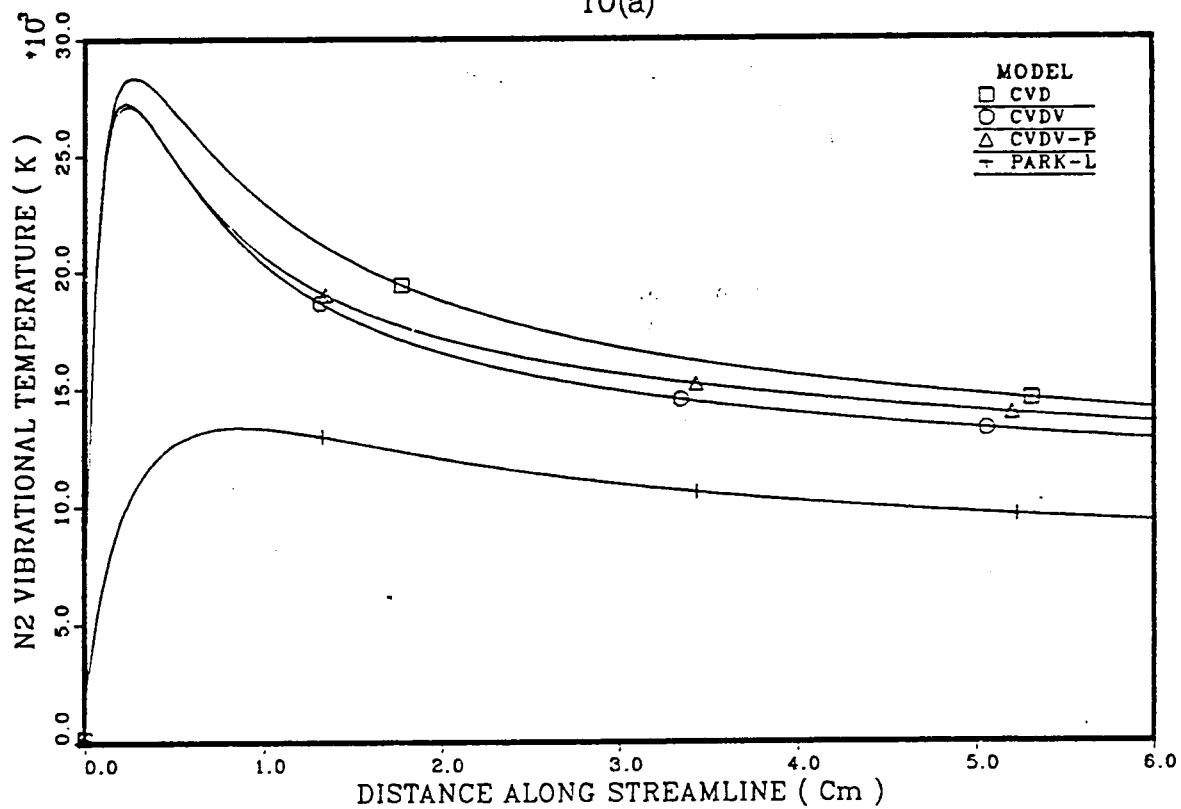


9(b)

FIGURES 9(a),9(b).PARK-L MODEL AT V=10 Km/s, RR1

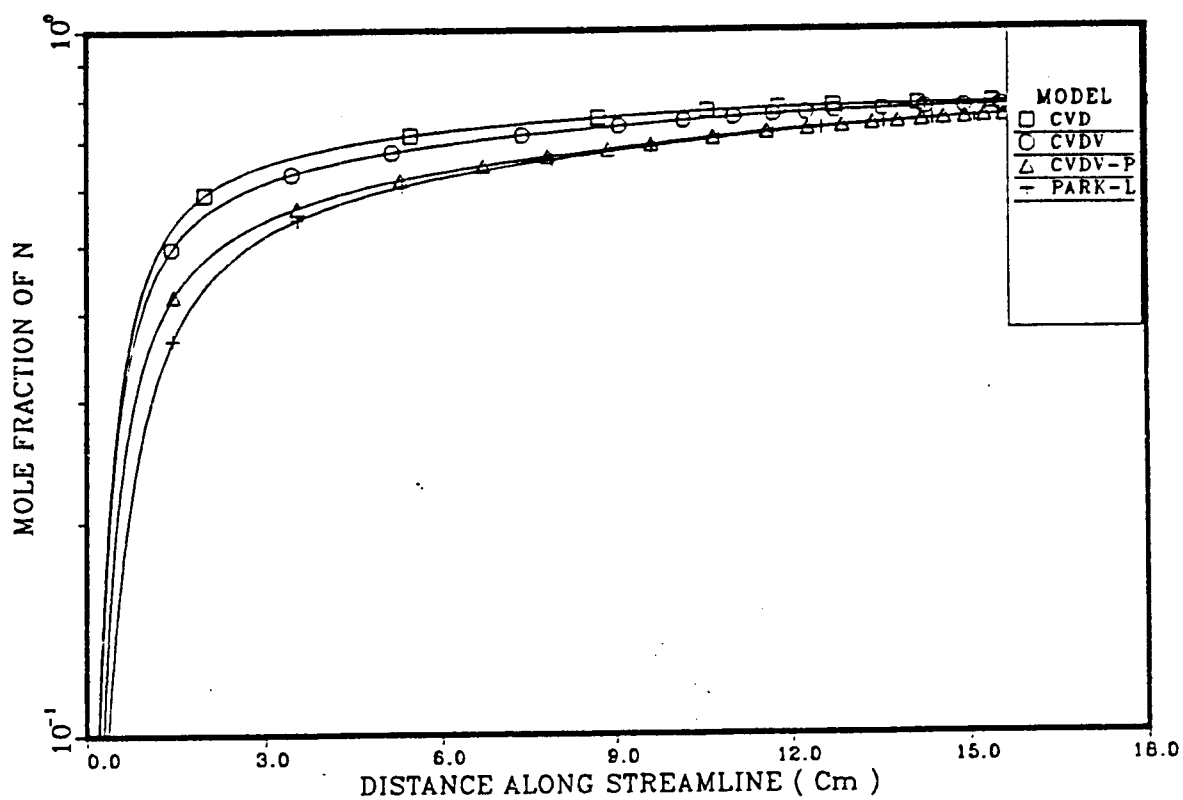


10(a)

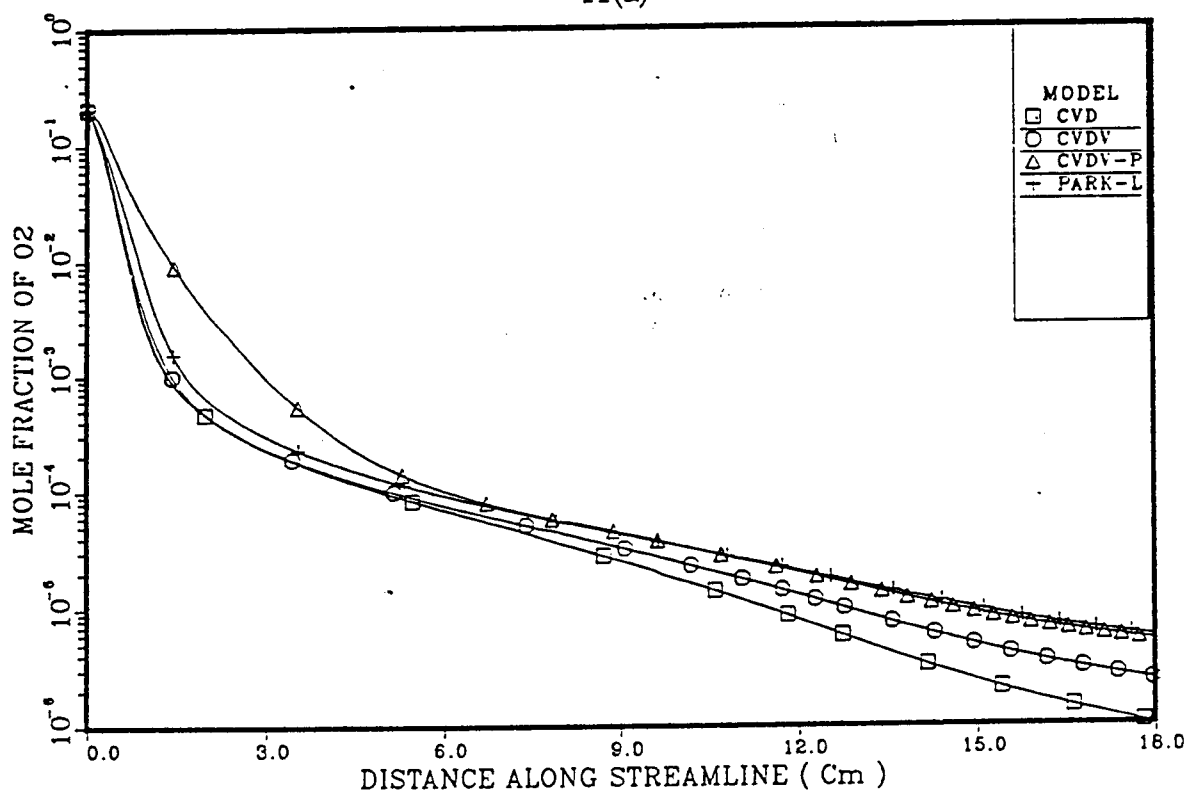


10(b)

FIGURES 10(a),10(b).PROFILES AT V=10 Km/s, RR1

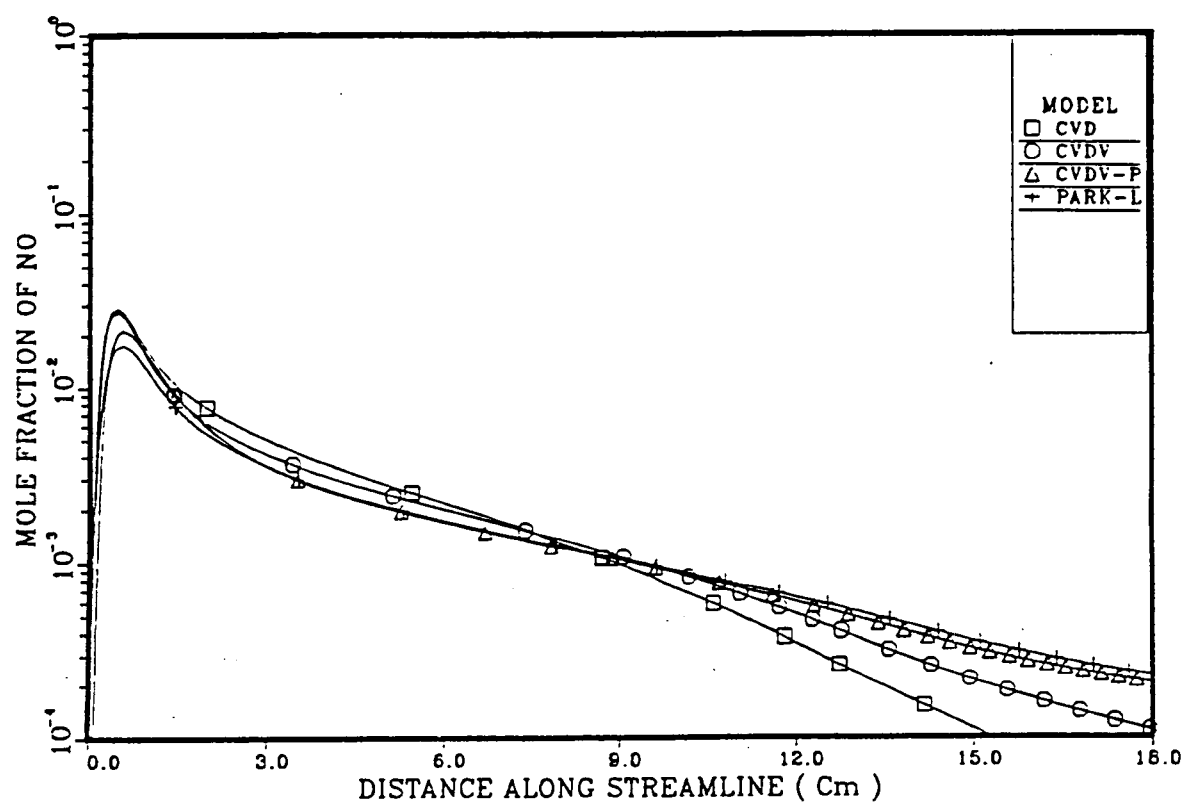


11(a)

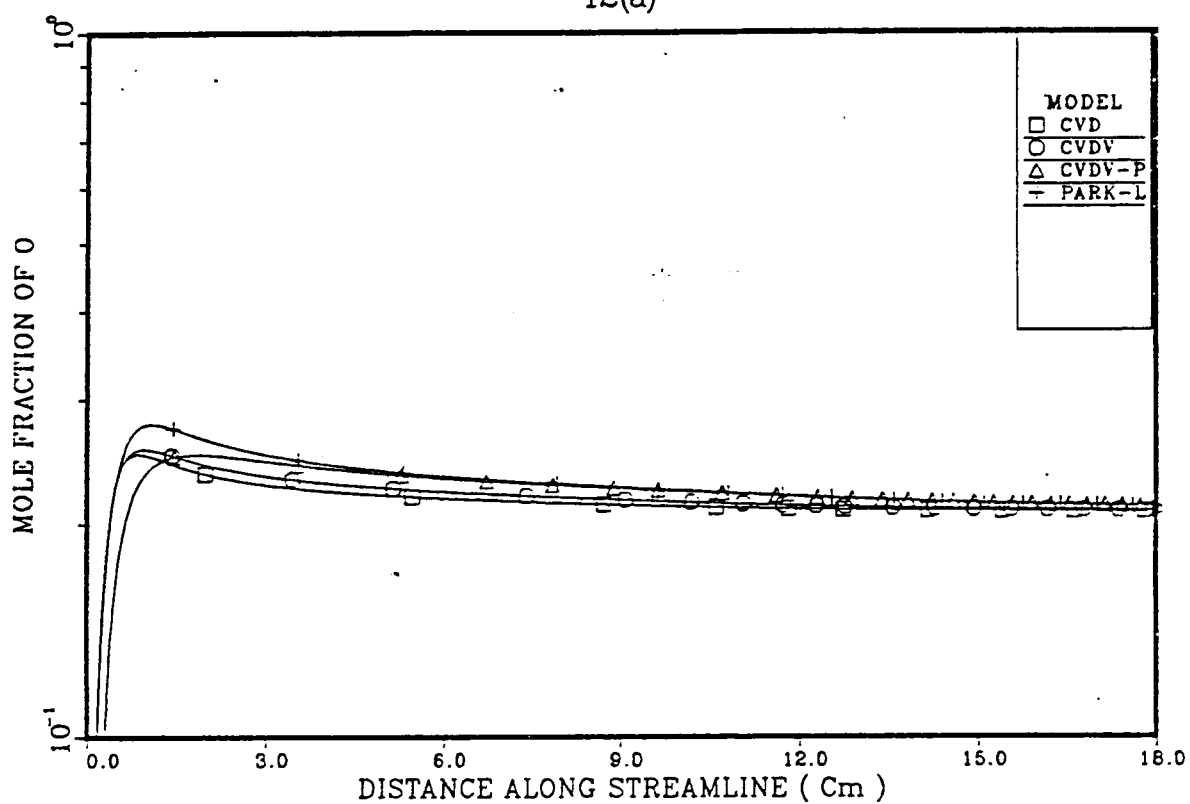


11(b)

FIGURES 11(a),11(b).PROFILES AT V=10 Km/s, RR1

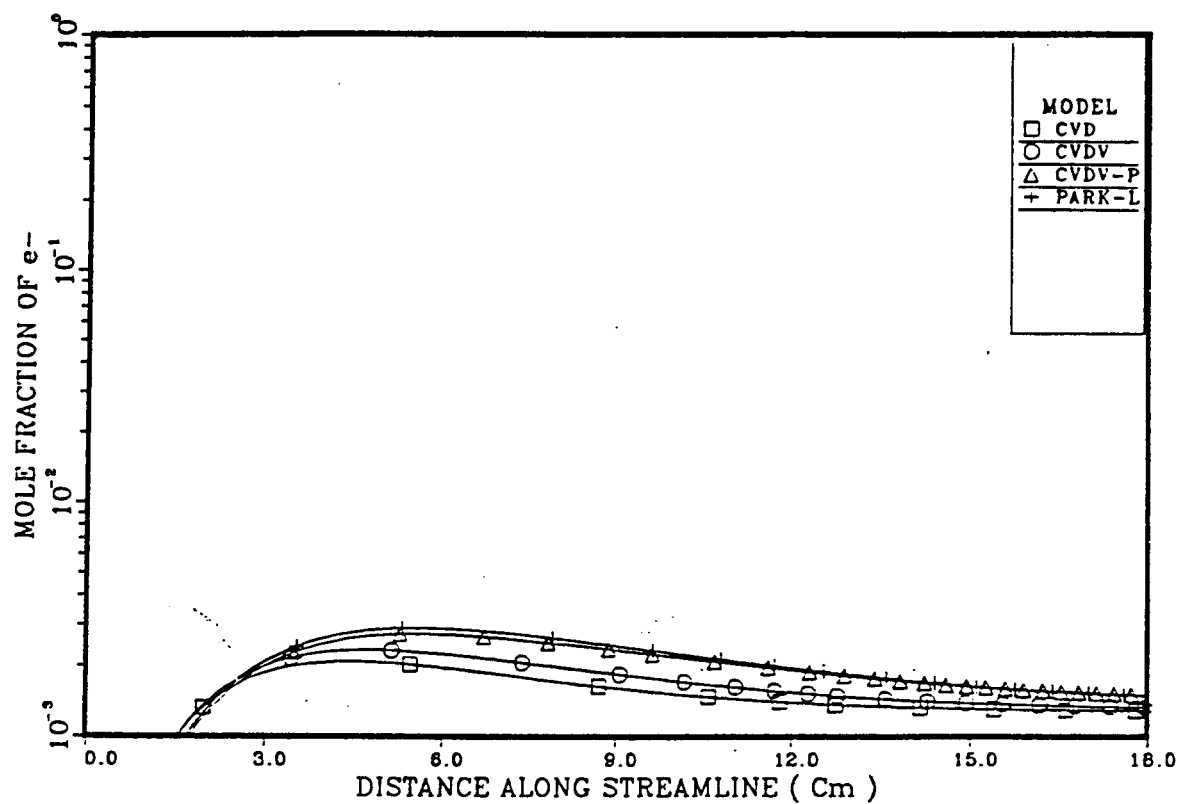


12(a)

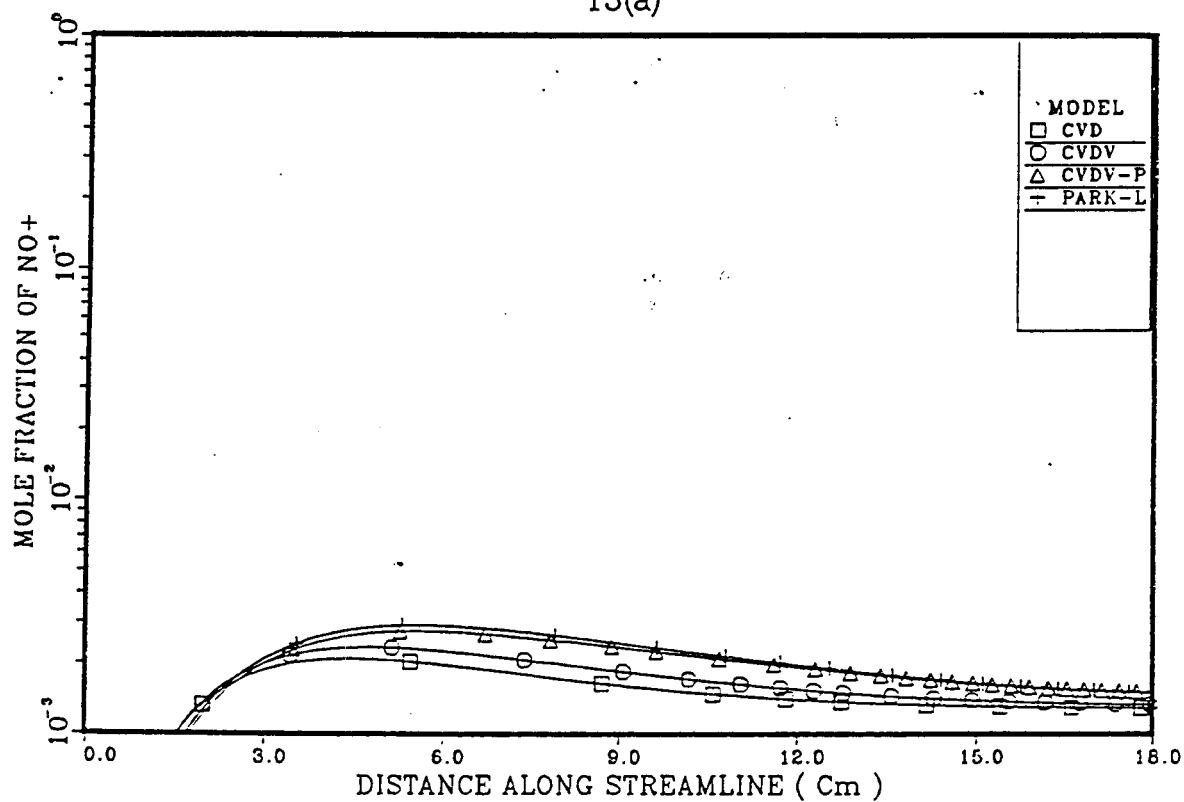


12(b)

FIGURES 12(a),12(b).PROFILES AT V=10 Km/s, RR1



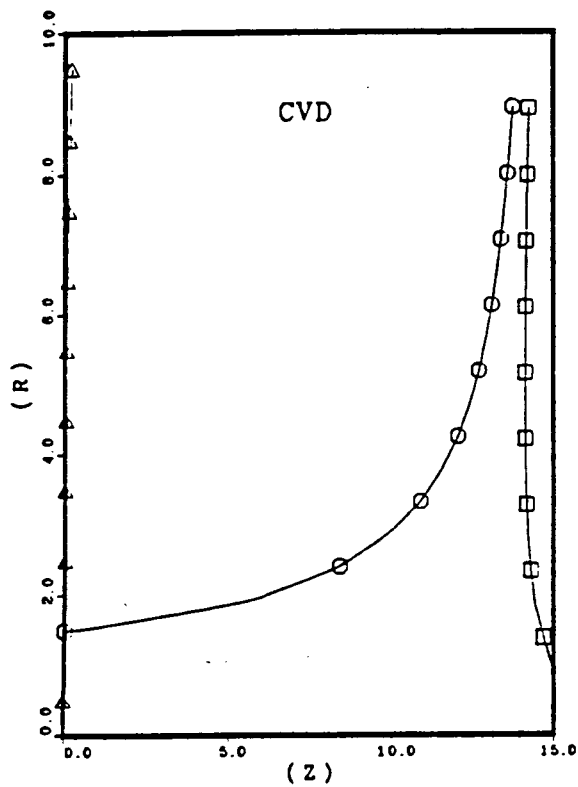
13(a)



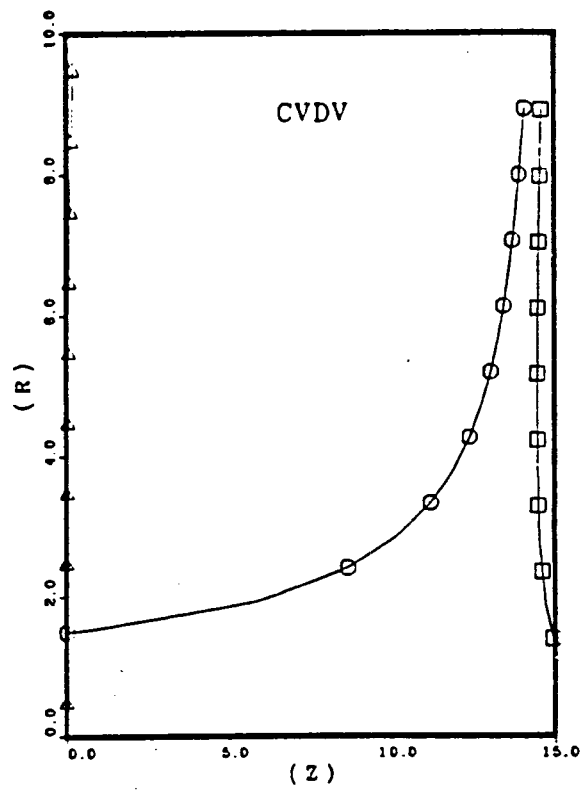
13(b)

FIGURES 13(a),13(b).PROFILES AT $V=10$ Km/s, RR1

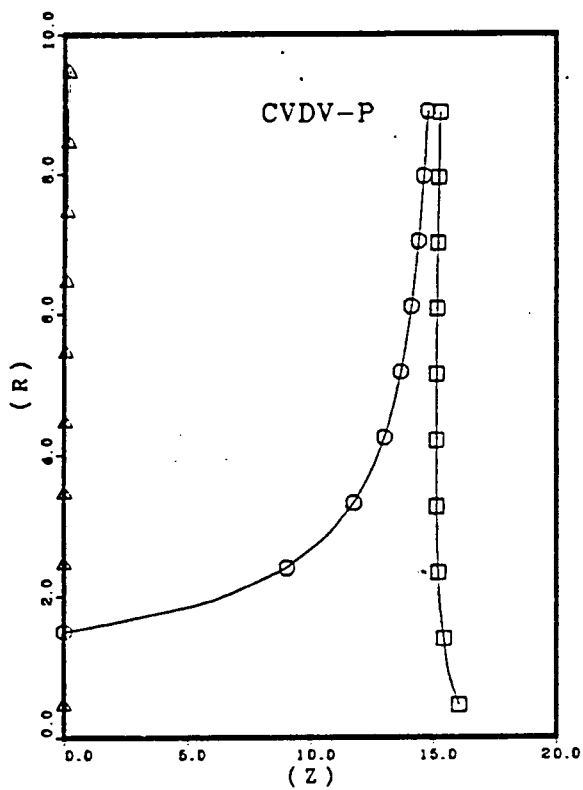
ORIGINAL PAGE IS
OF POOR QUALITY



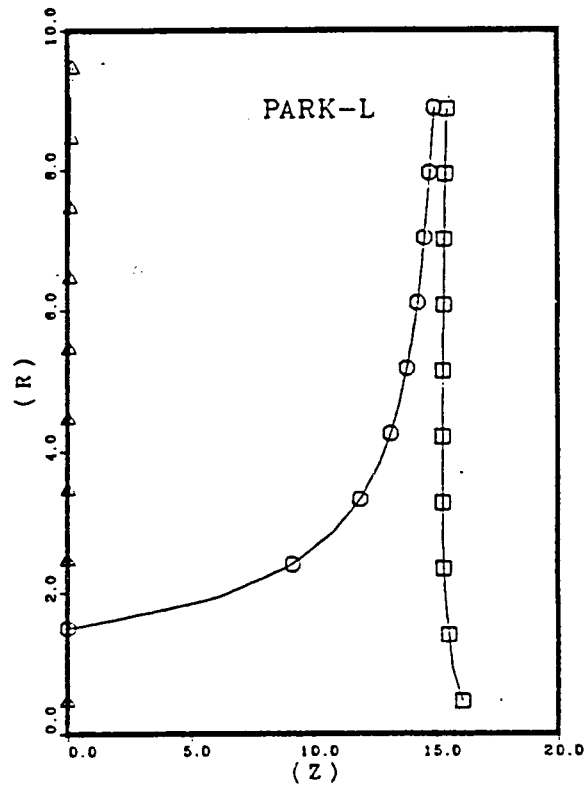
14(a)



14(b)

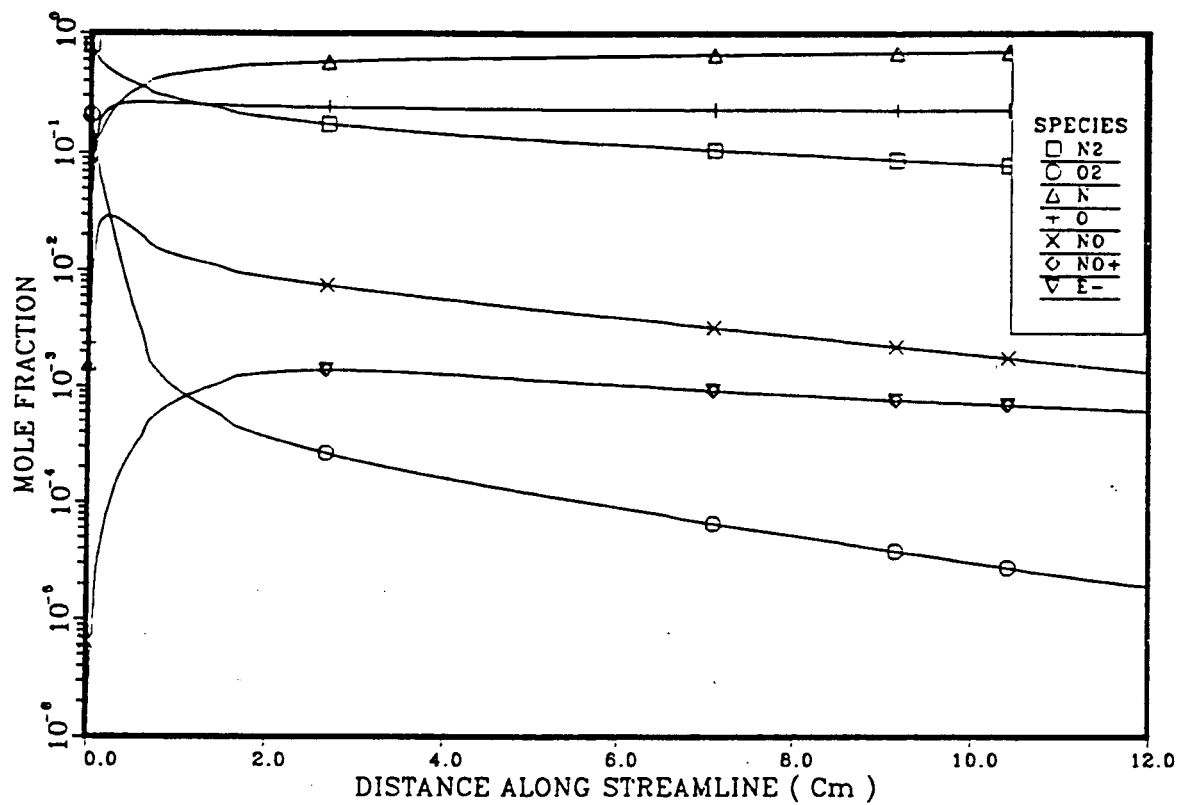


14(c)

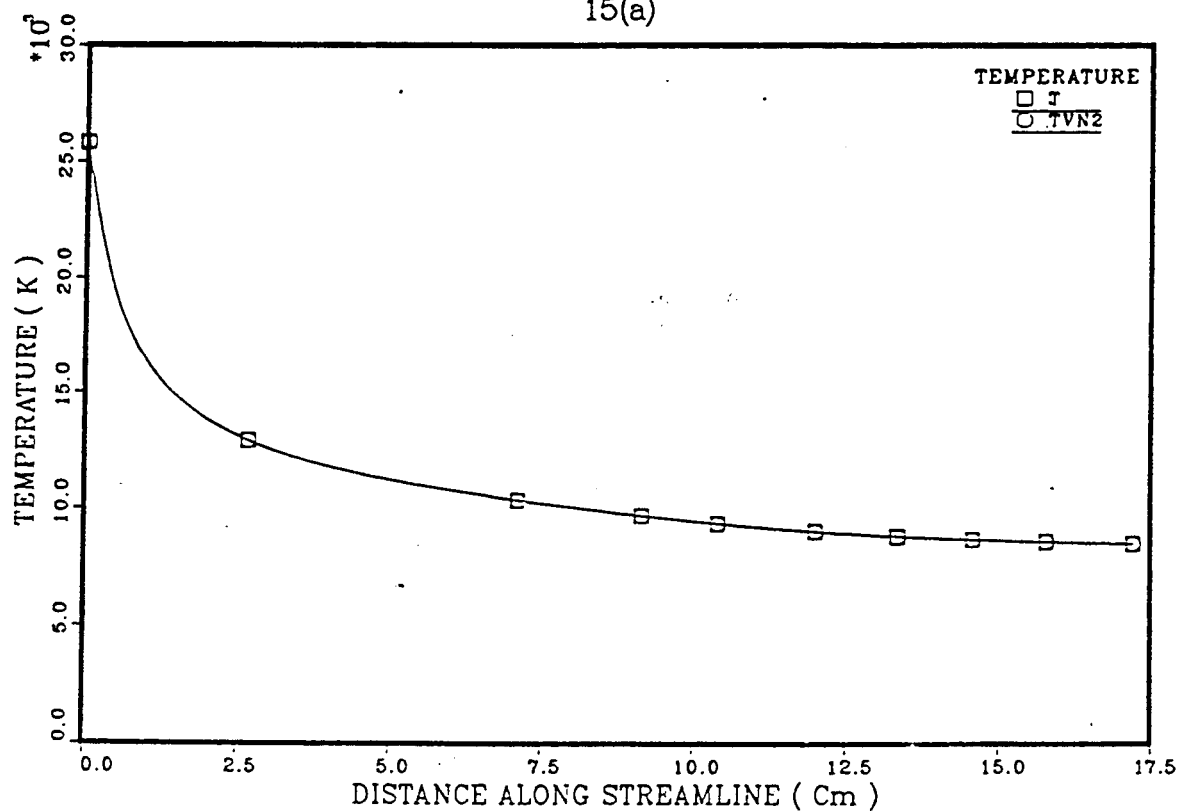


14(d)

FIGURES 14(a),14(b),14(c),14(d).COORD,V=10 Km/s, RR1

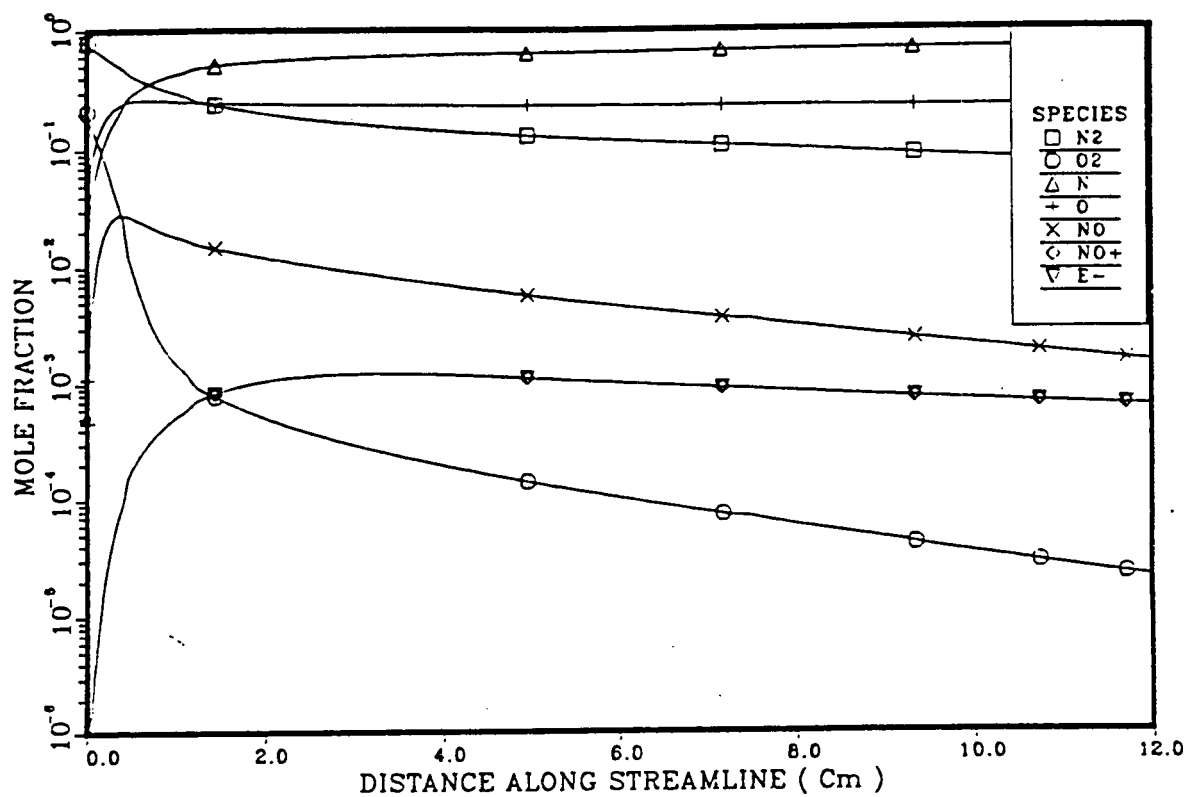


15(a)

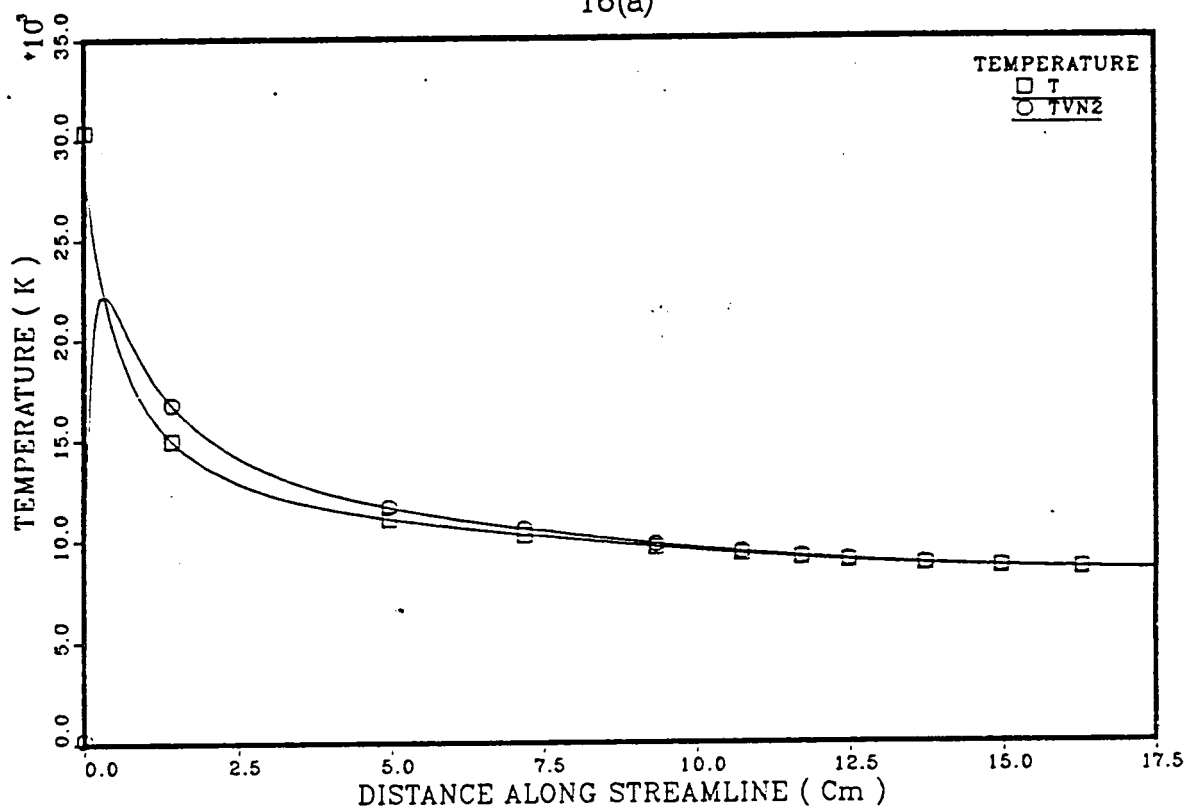


15(b)

FIGURES 15(a),15(b).VEQ MODEL AT V=8.9 Km/s, RR1

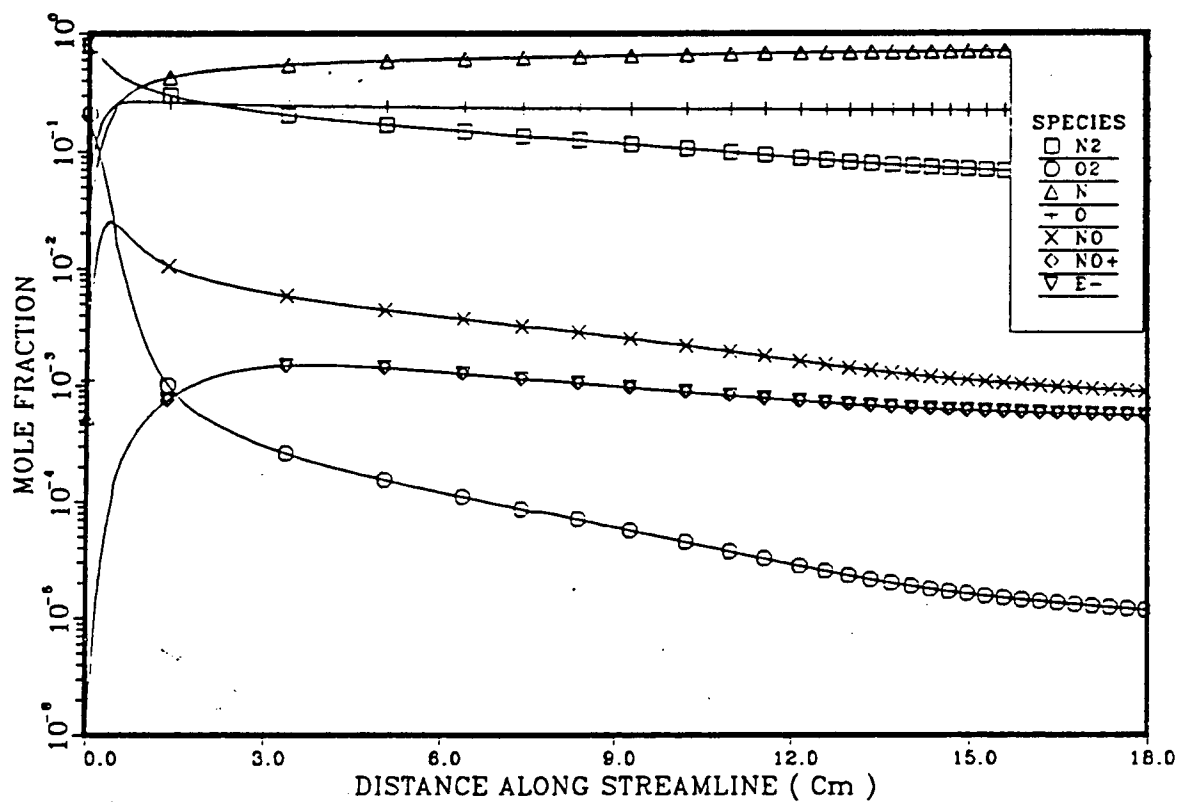


16(a)

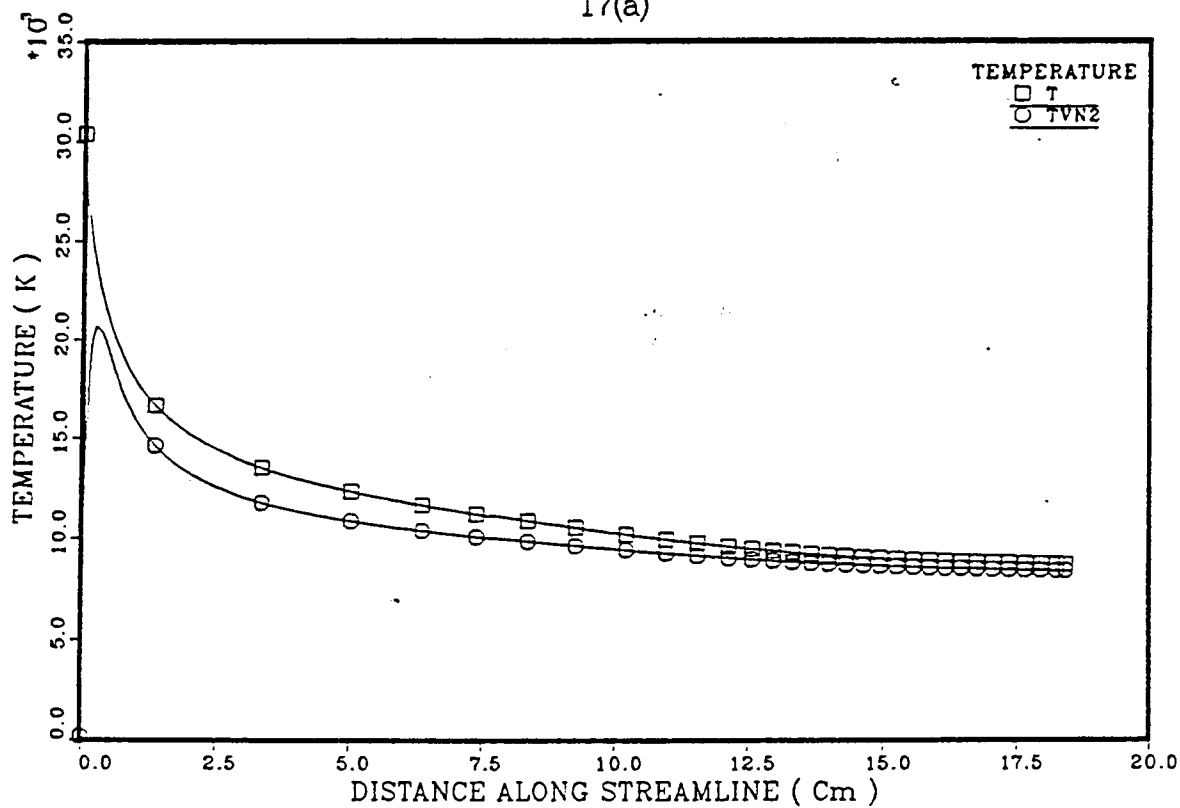


16(b)

FIGURES 16(a),16(b).CVD MODEL AT V=8.9 Km/s, RR1

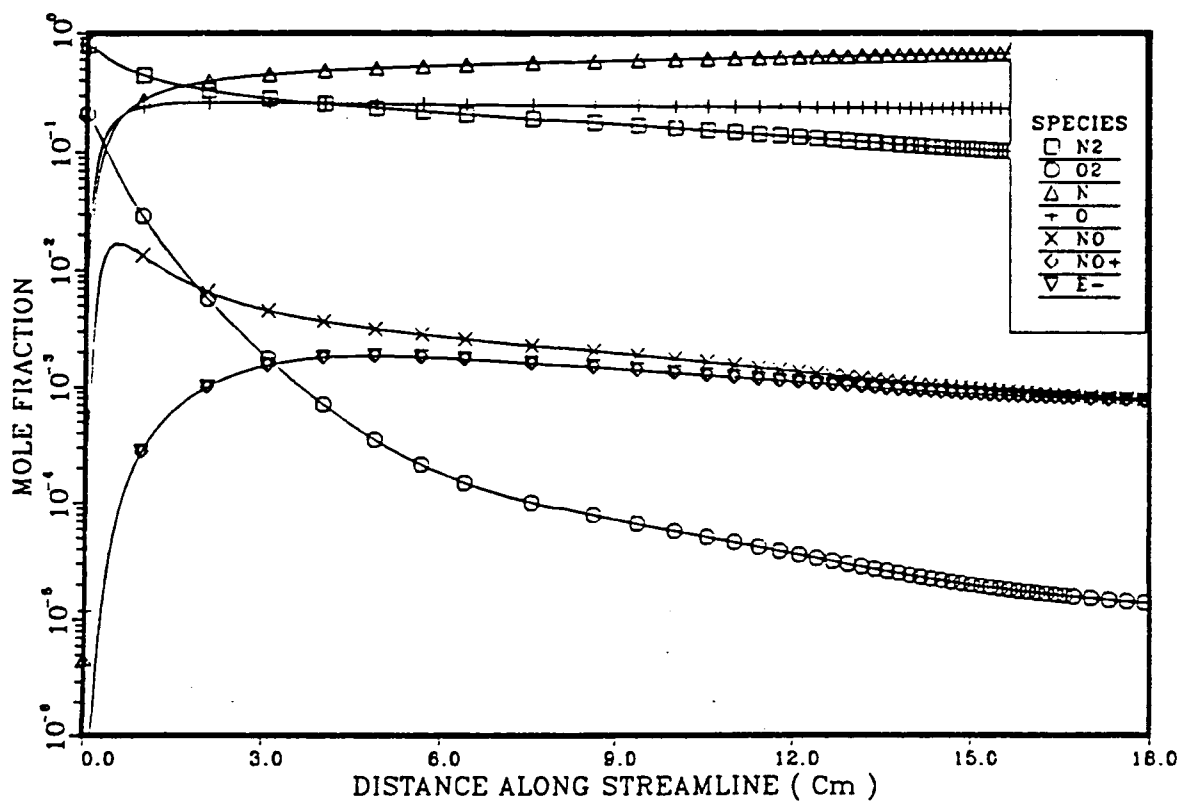


17(a)

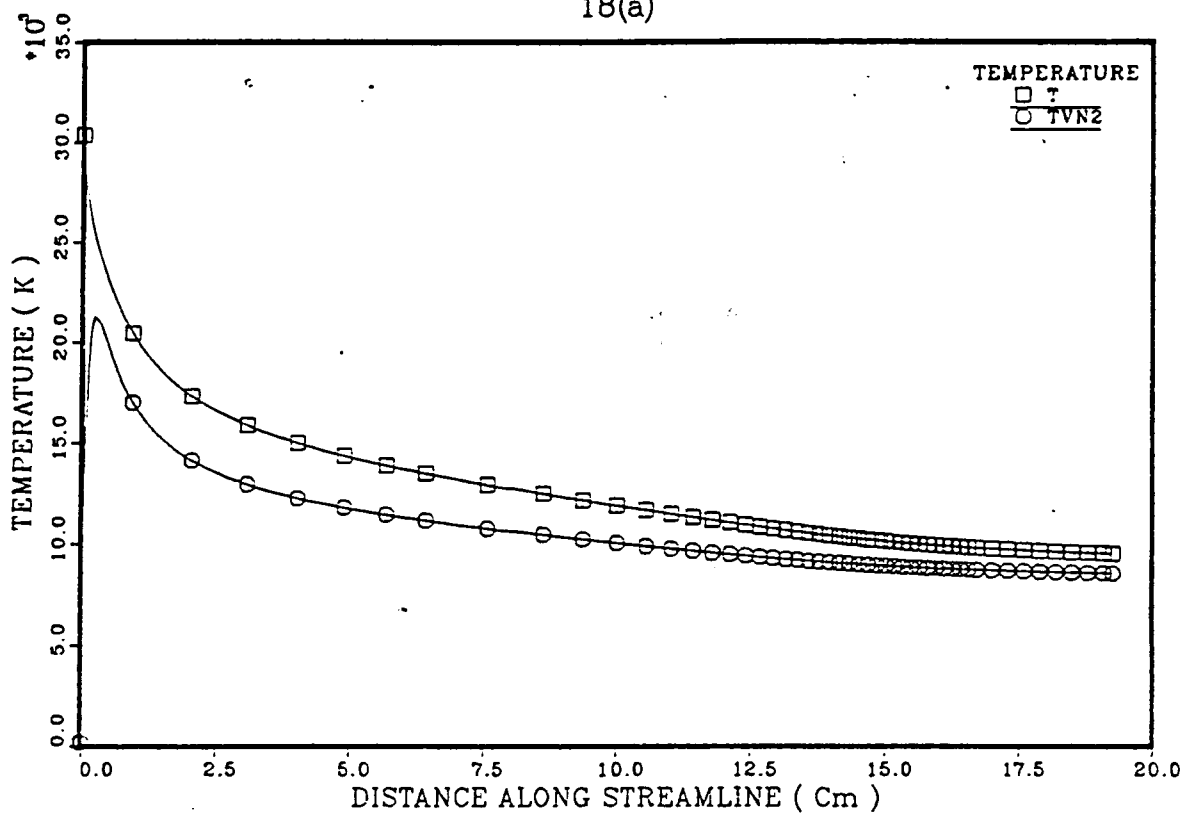


17(b)

FIGURES 17(a),17(b).CVDV MODEL AT V=8.9 Km/s, RR1

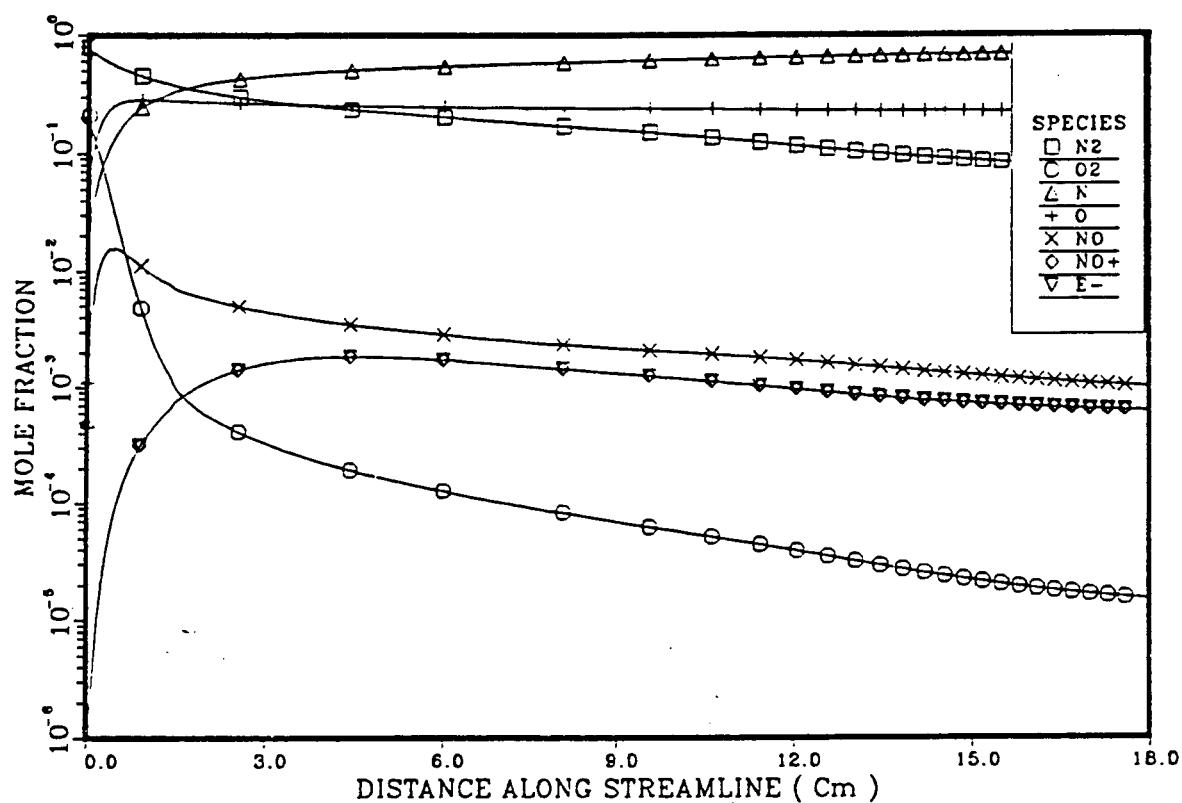


18(a)

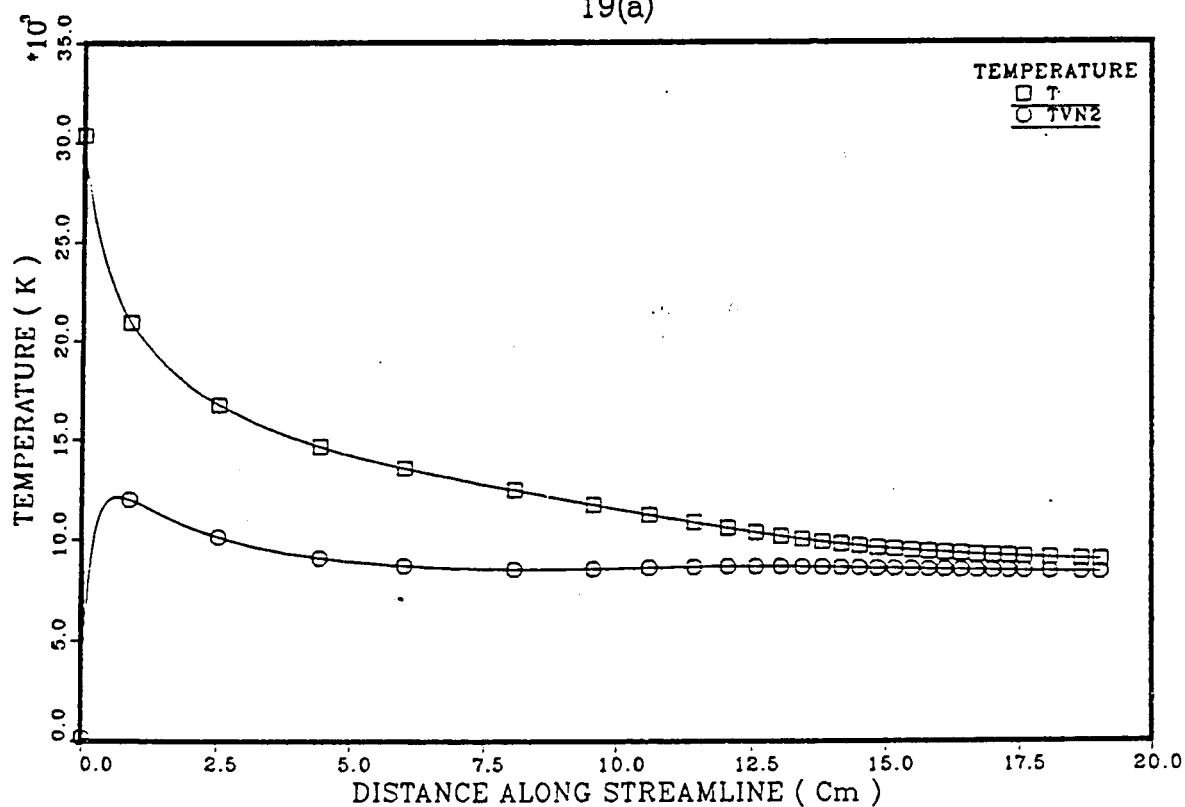


18(b)

FIGURES 18(a), 18(b). CVDV-P MODEL AT $V=8.9$ Km/s, RR1

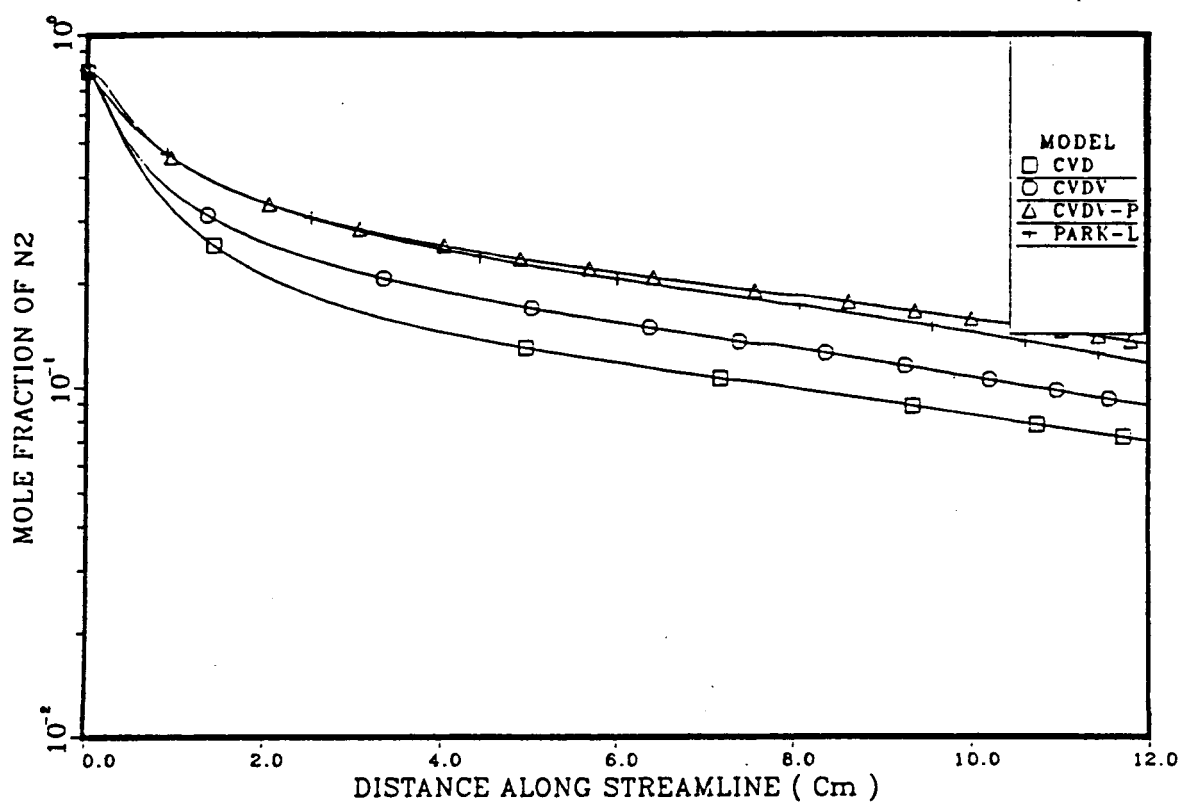


19(a)

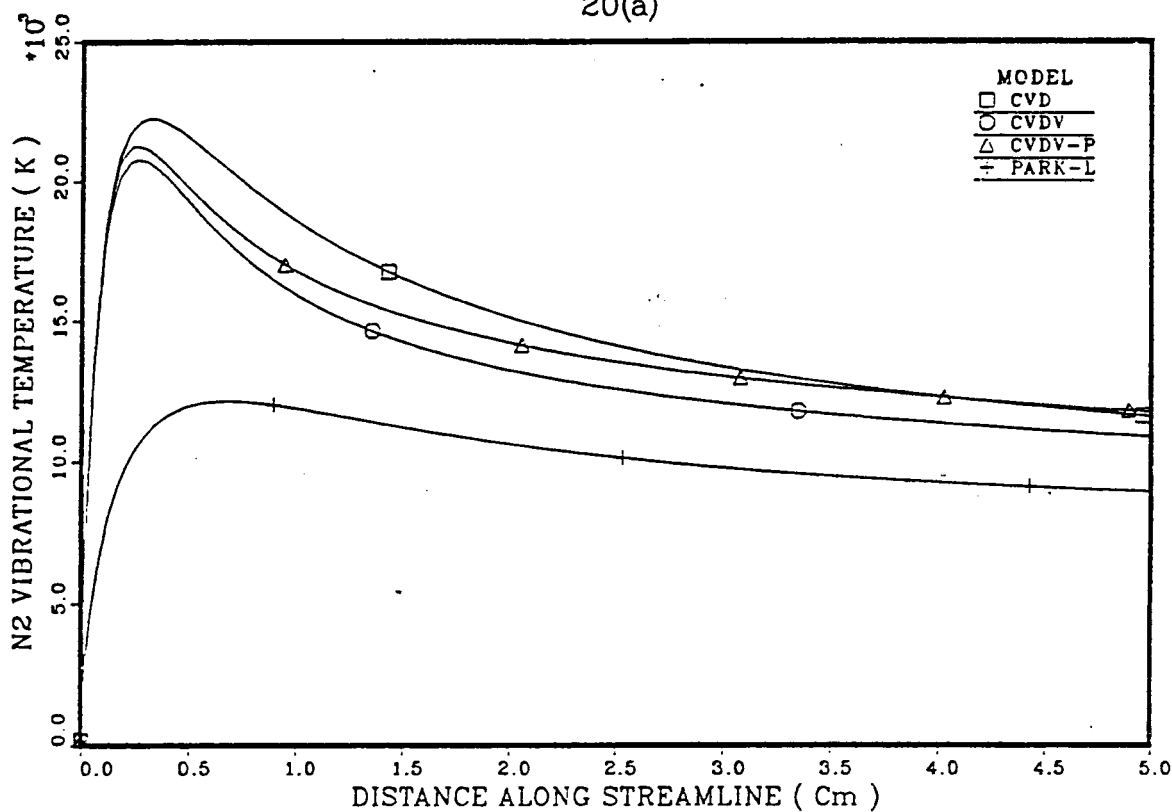


19(b)

FIGURES 19(a),19(b).PARK-L MODEL AT V=8.9 Km/s, RR1

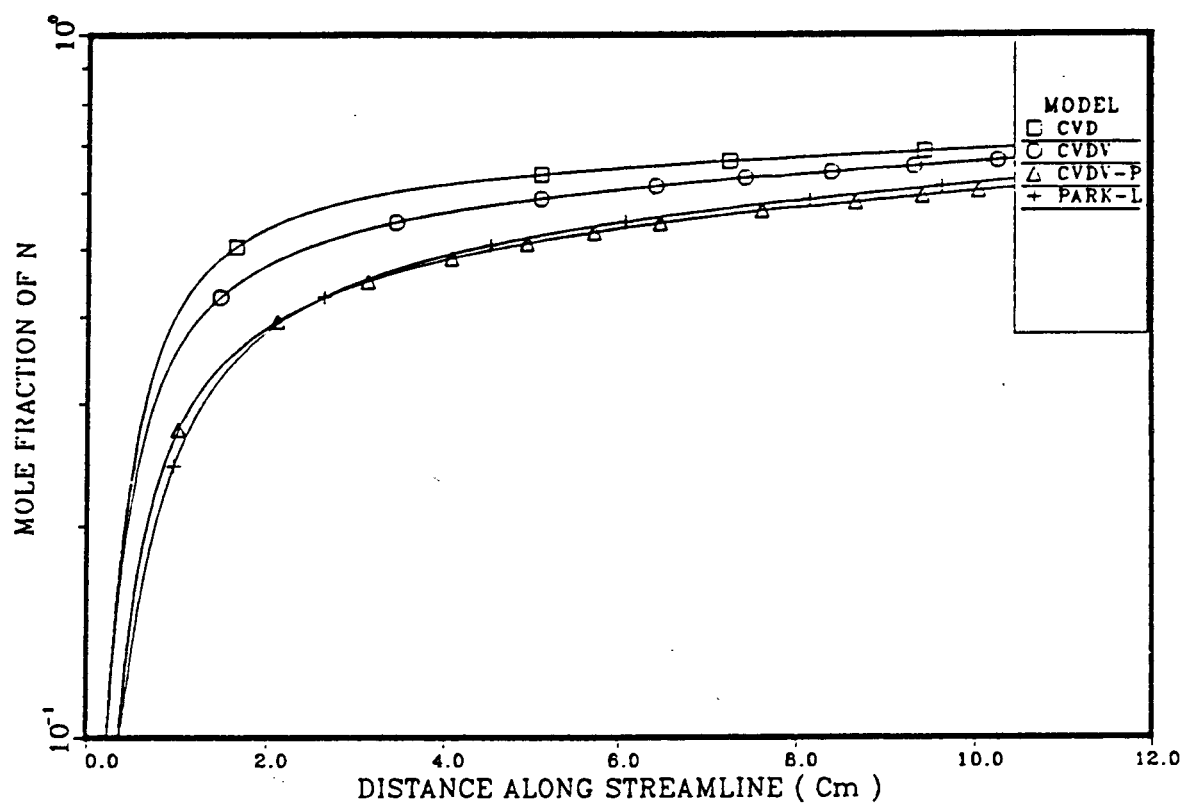


20(a)

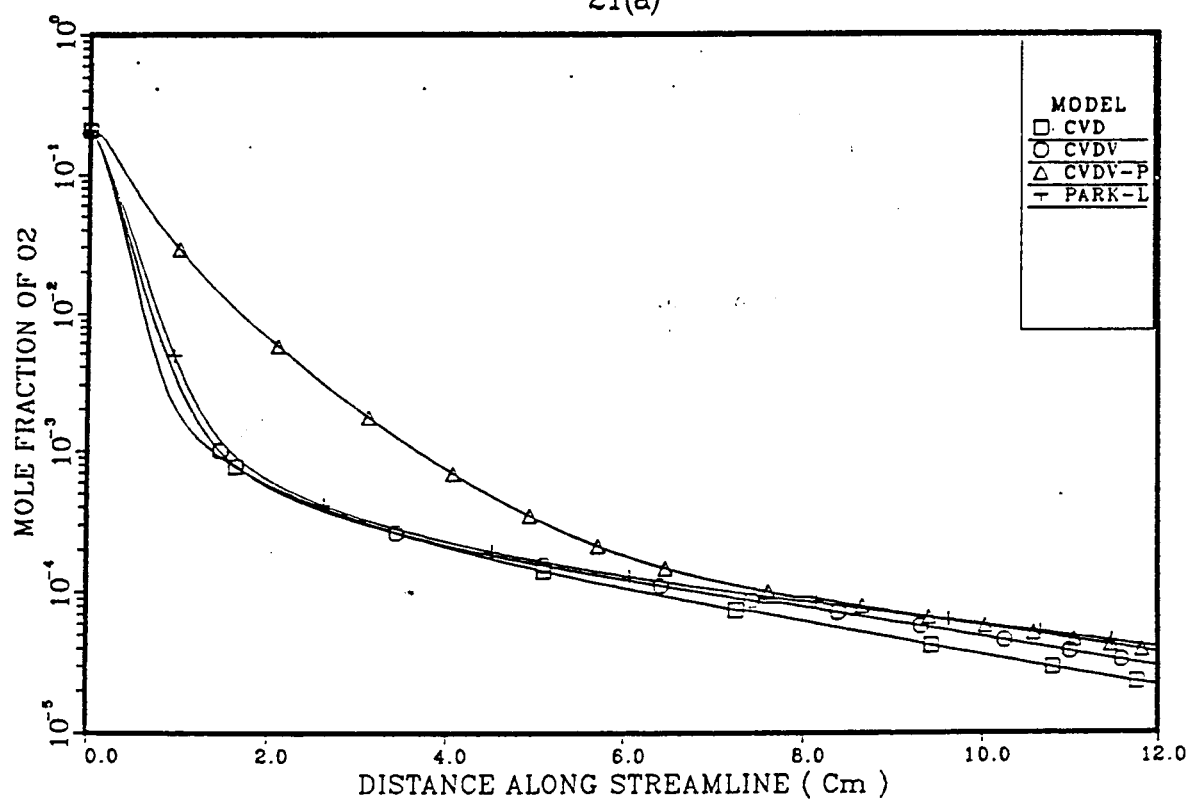


20(b)

FIGURES 20(a),20(b).PROFILES AT V=8.9 Km/s, RR1

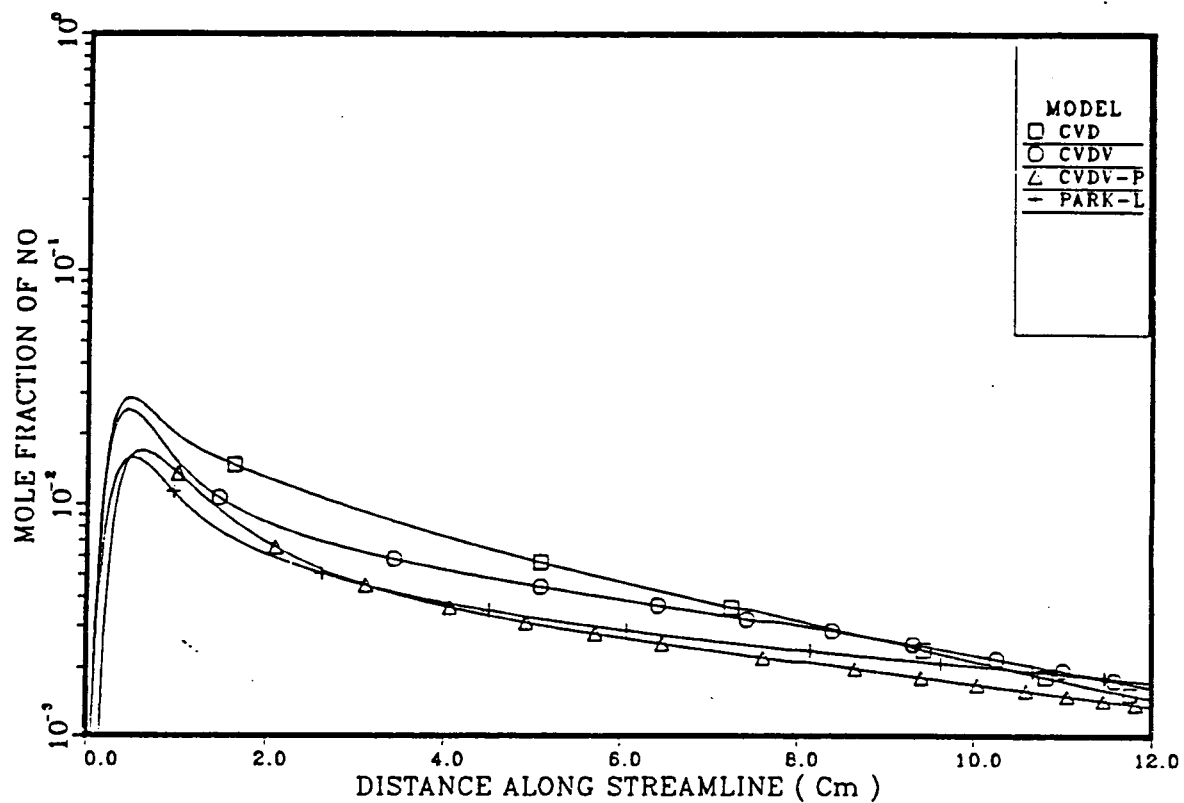


21(a)

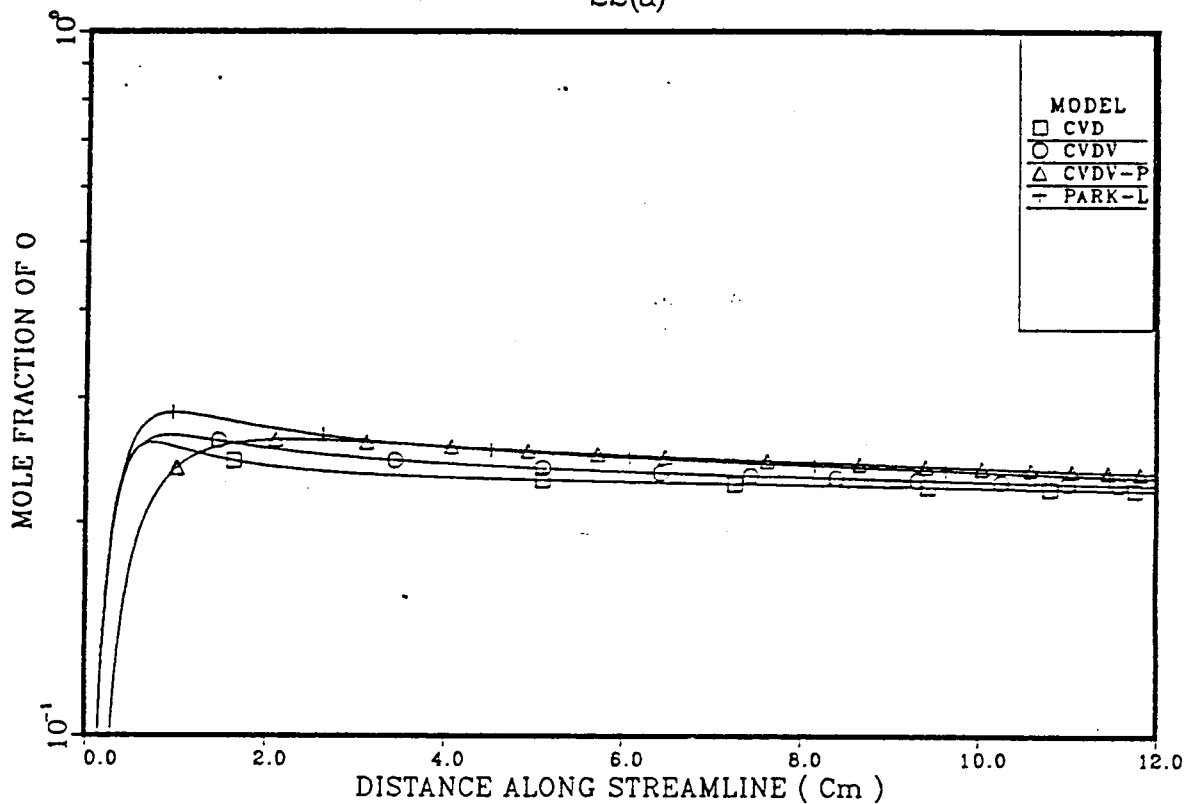


21(b)

FIGURES 21(a),21(b).PROFILES AT V=8.9 Km/s, RR1

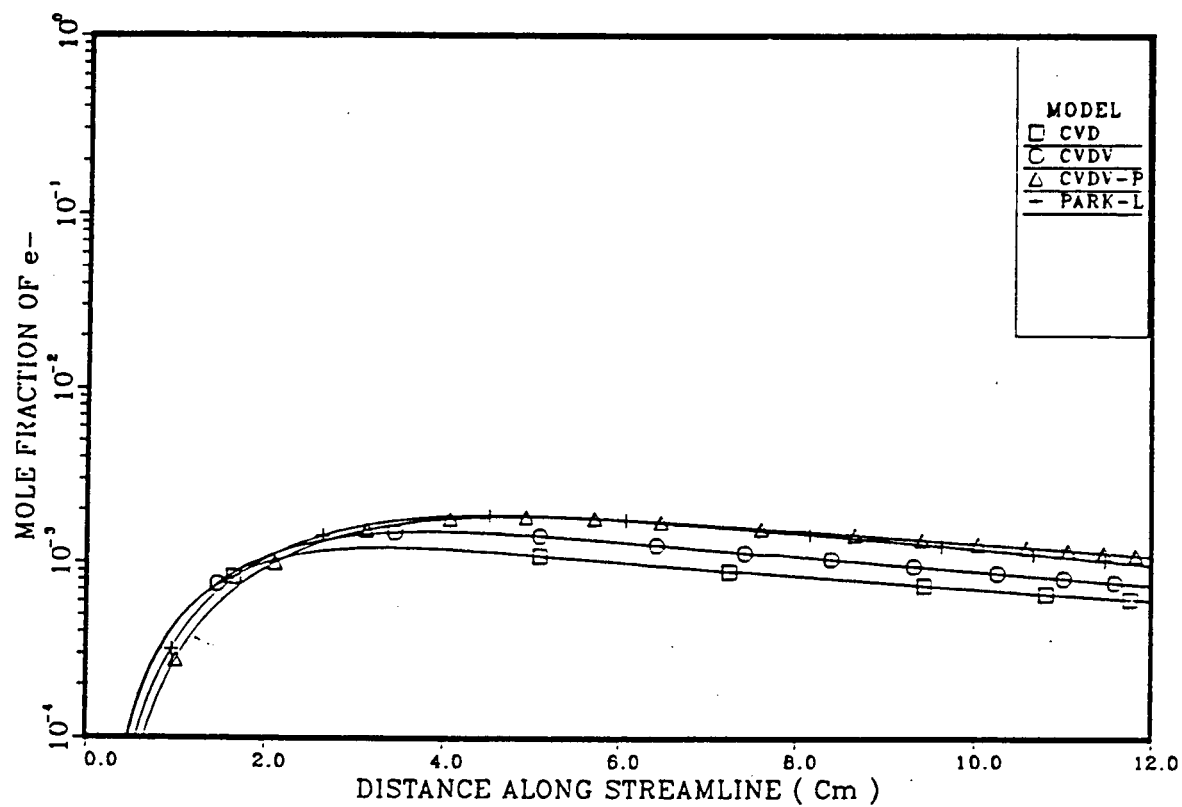


22(a)

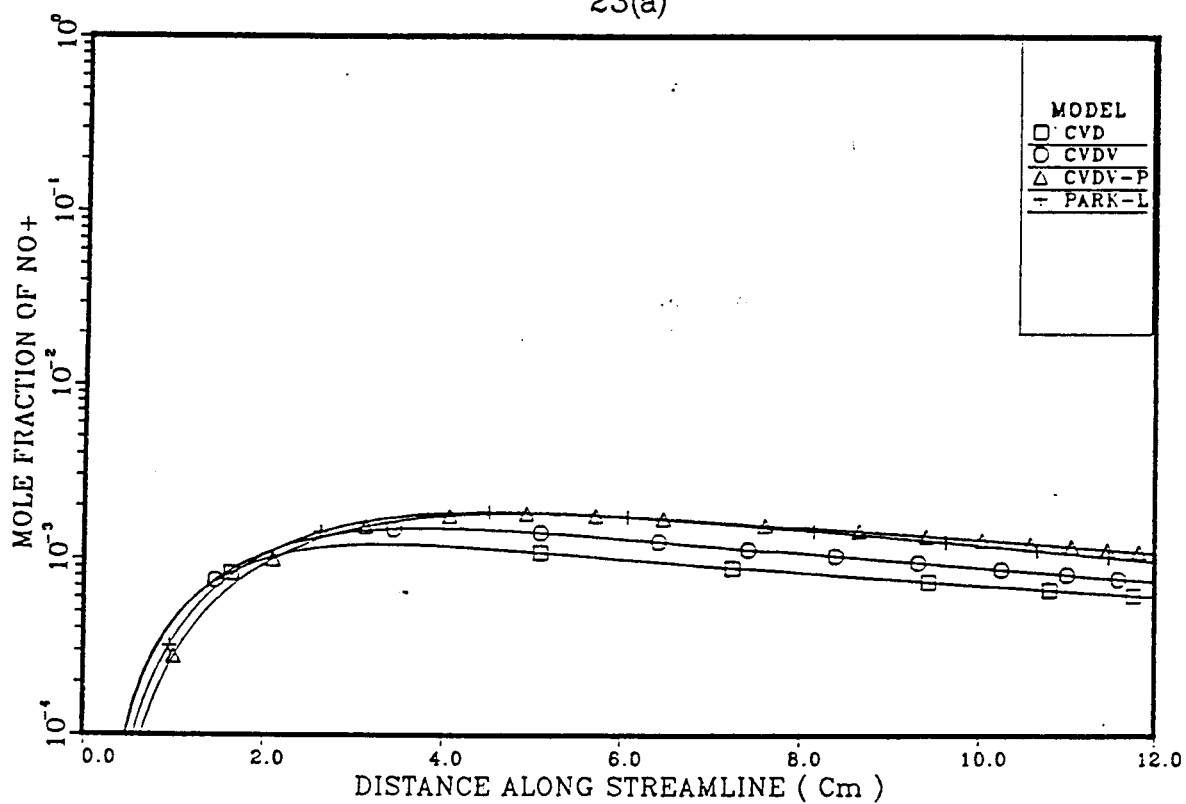


22(b)

FIGURES 22(a),22(b).PROFILES AT V=8.9 Km/s, RR1

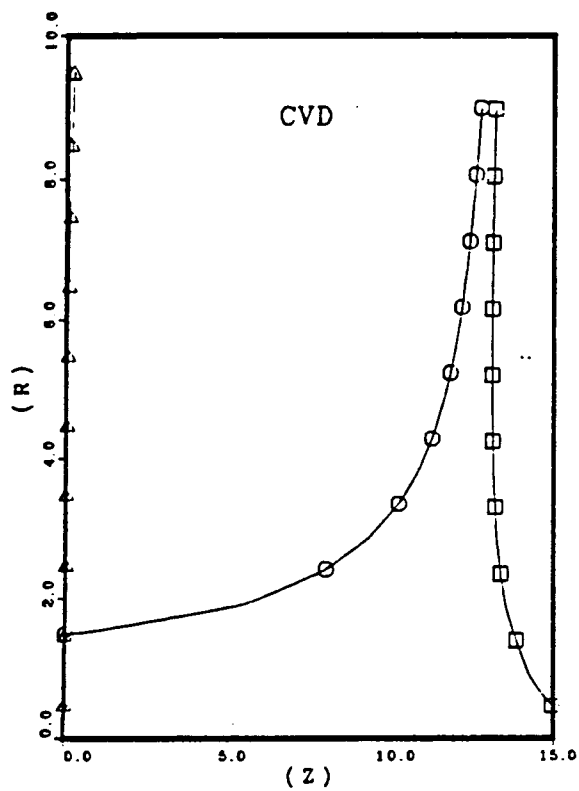


23(a)

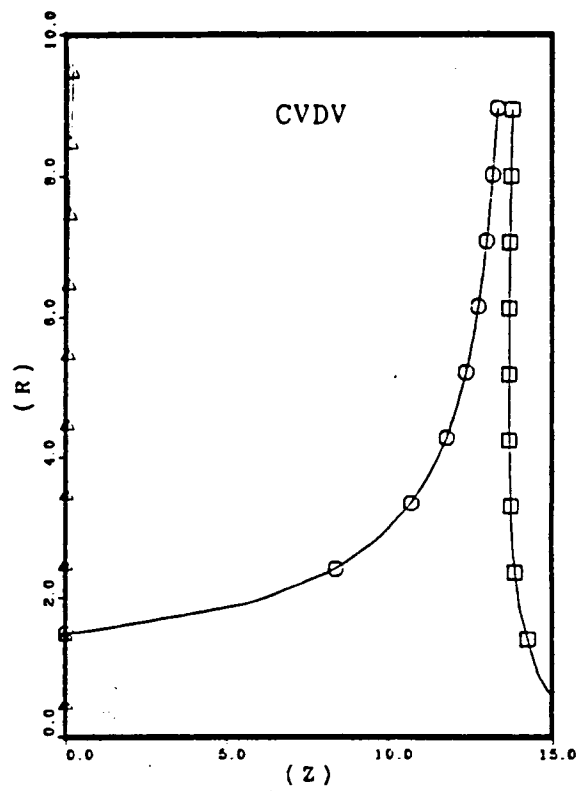


23(b)

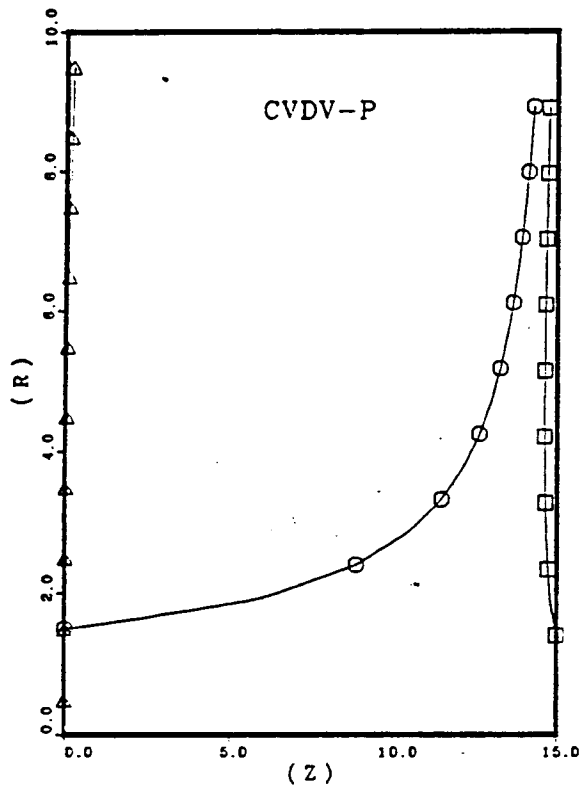
FIGURES 23(a),23(b).PROFILES AT $V=8.9$ Km/s, RR1



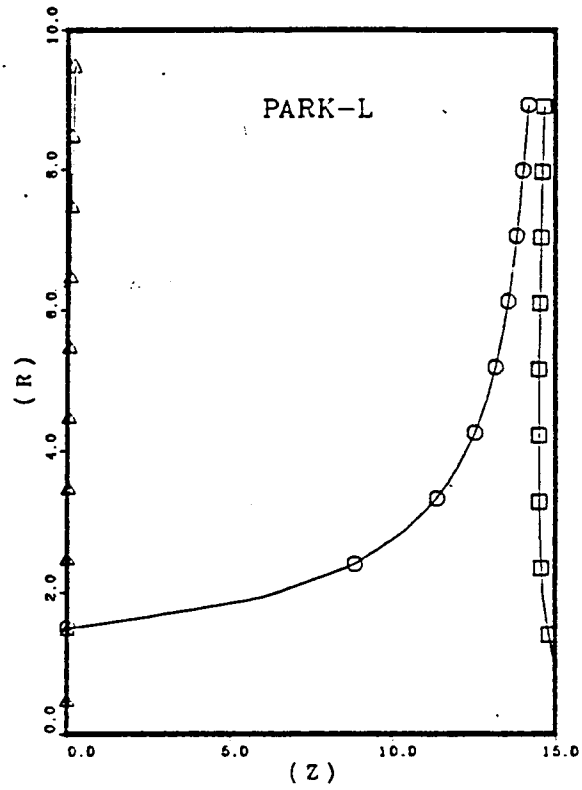
24(a)



24(b)

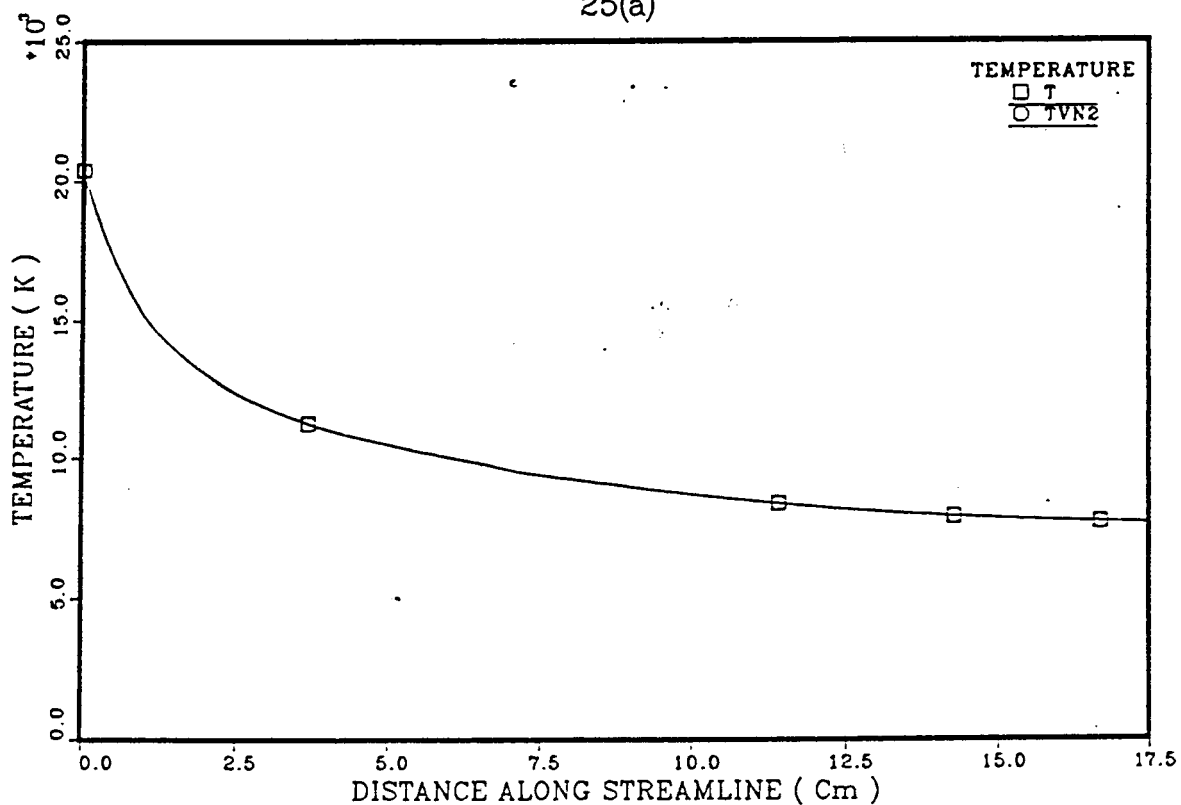
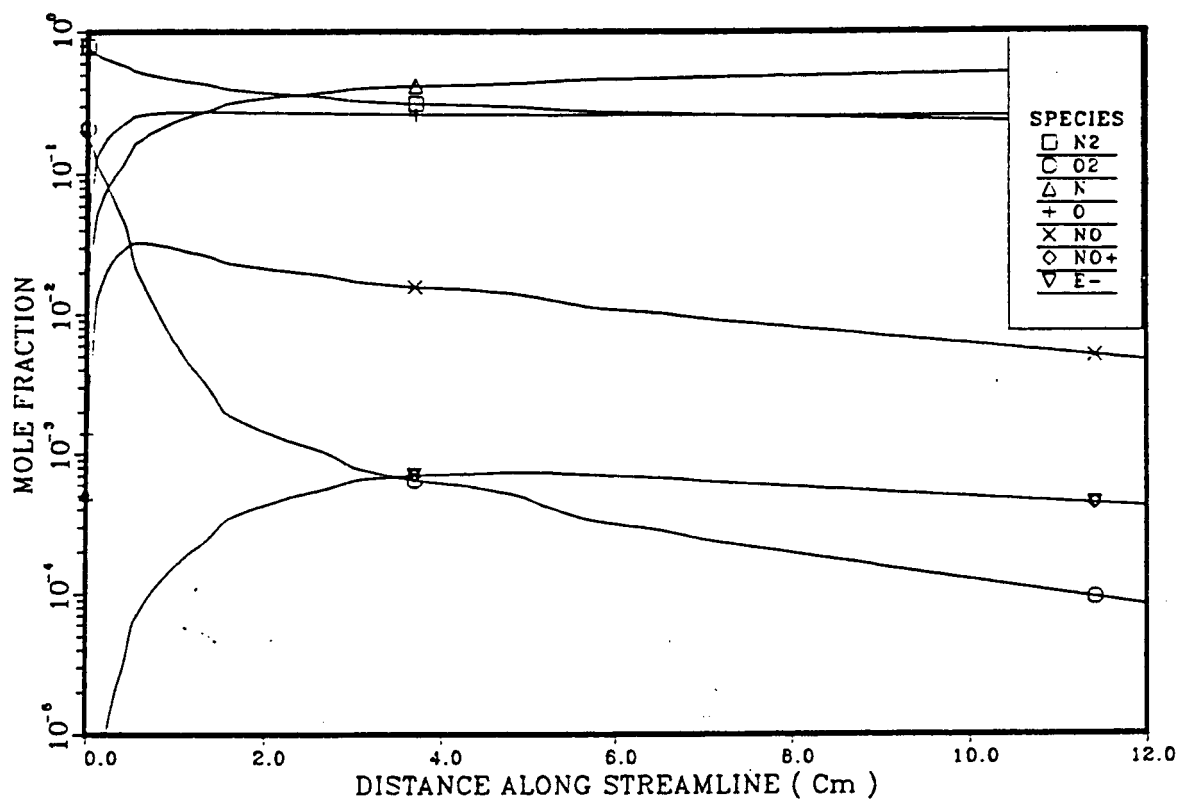


24(c)

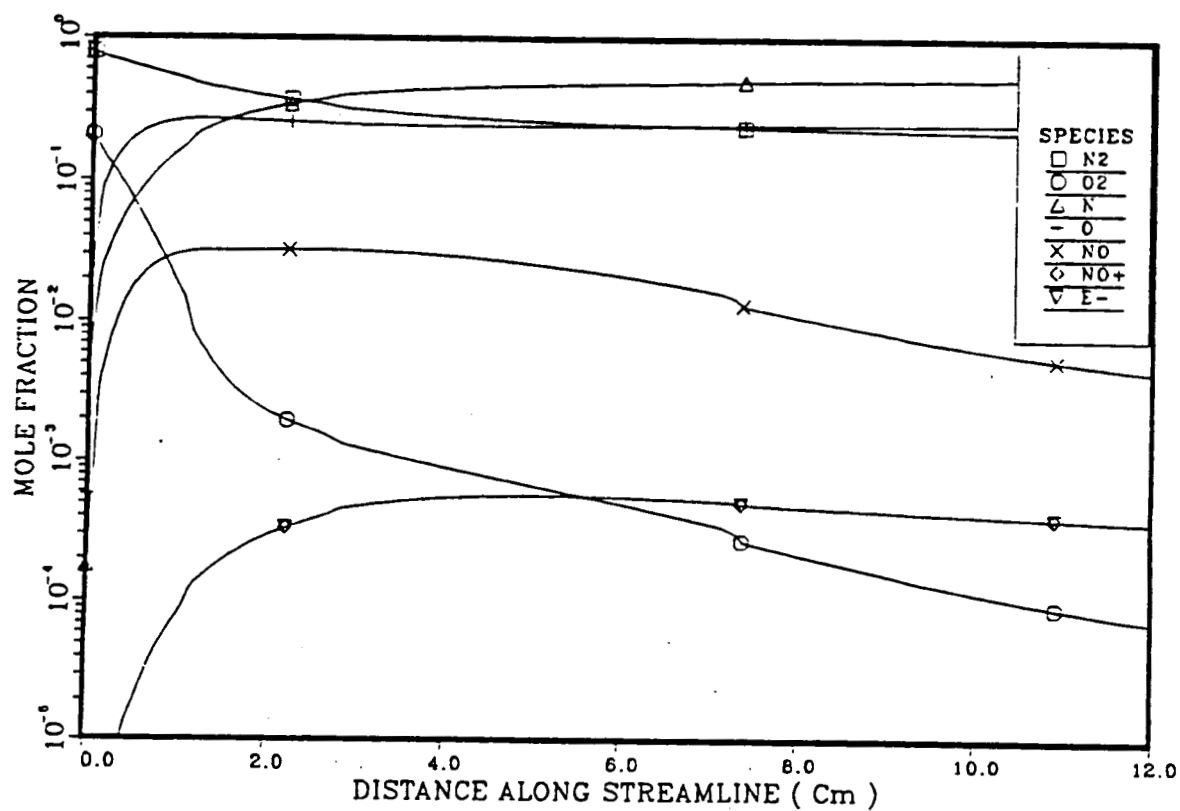


24(d)

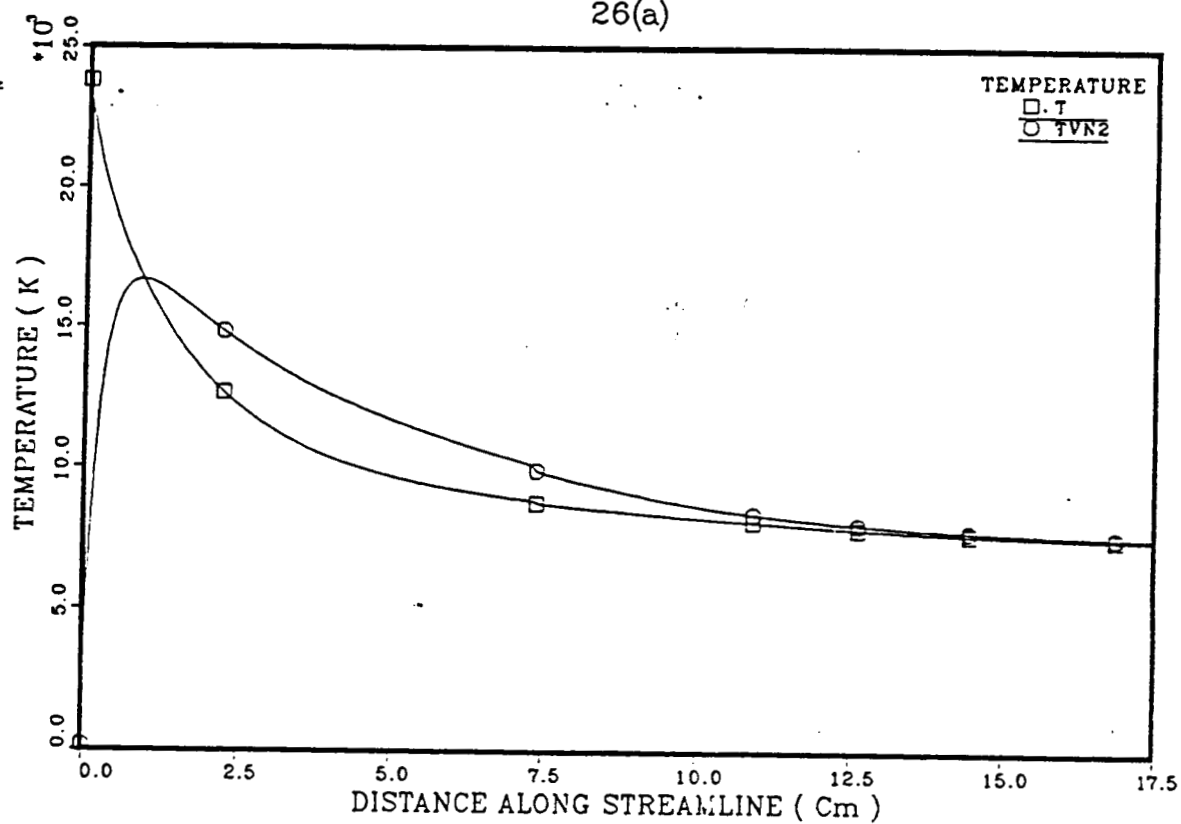
FIGURES 24(a),24(b),24(c),24(d).COORD,V=8.9 Km/s, RR1



FIGURES 25(a),25(b).VEQ MODEL AT V=7.7 Km/s, RR1

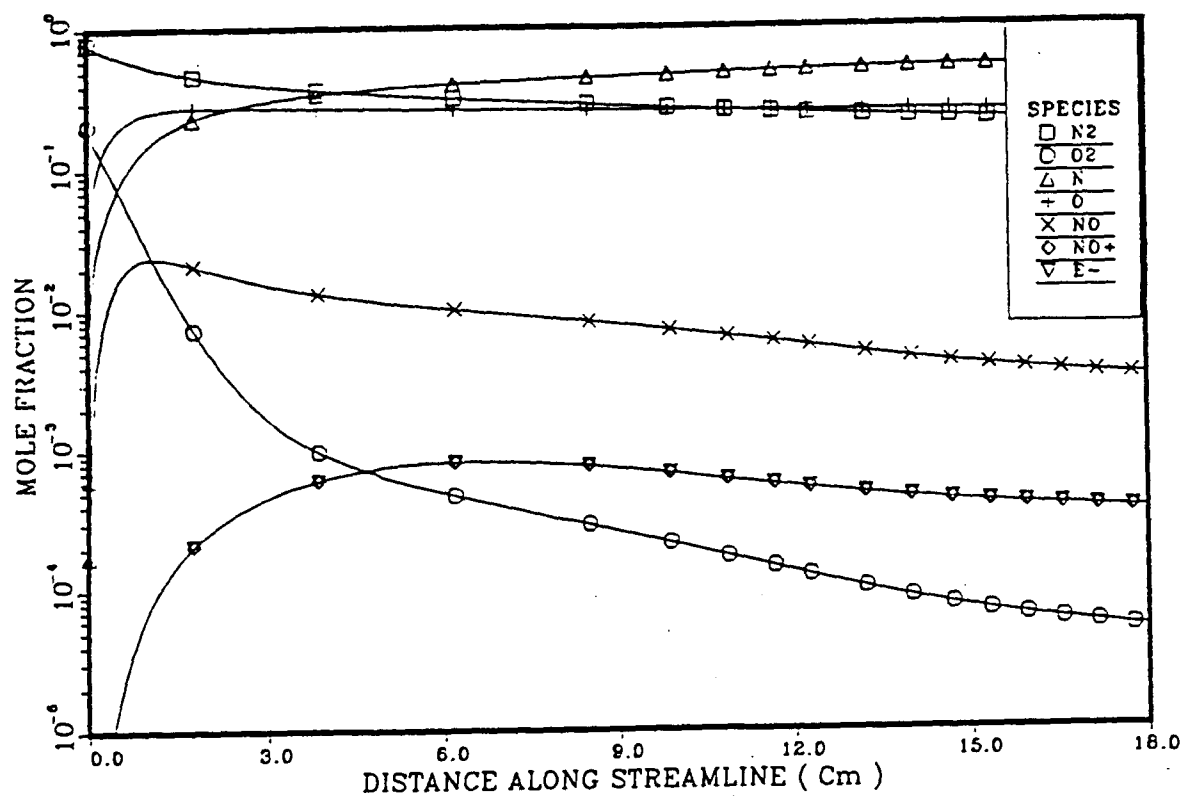


26(a)

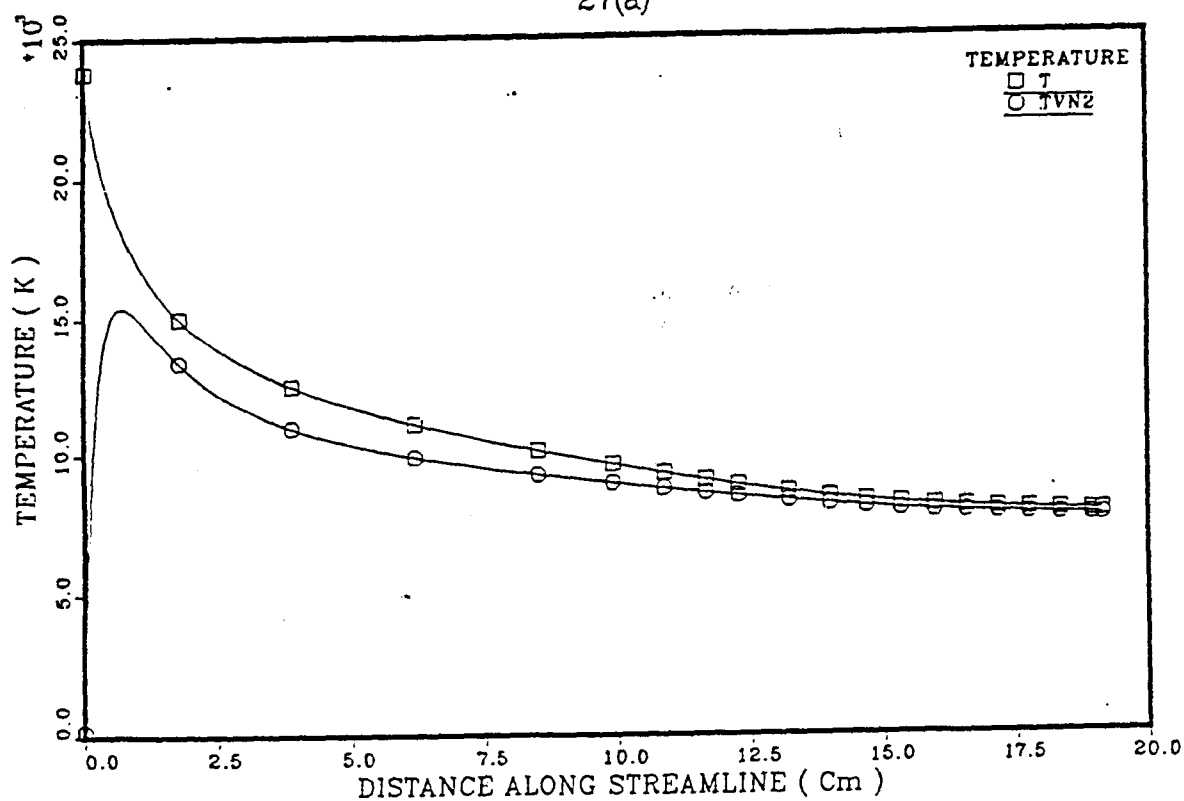


26(b)

FIGURES 26(a),26(b).CVD MODEL AT $V=7.7$ Km/s, RR1



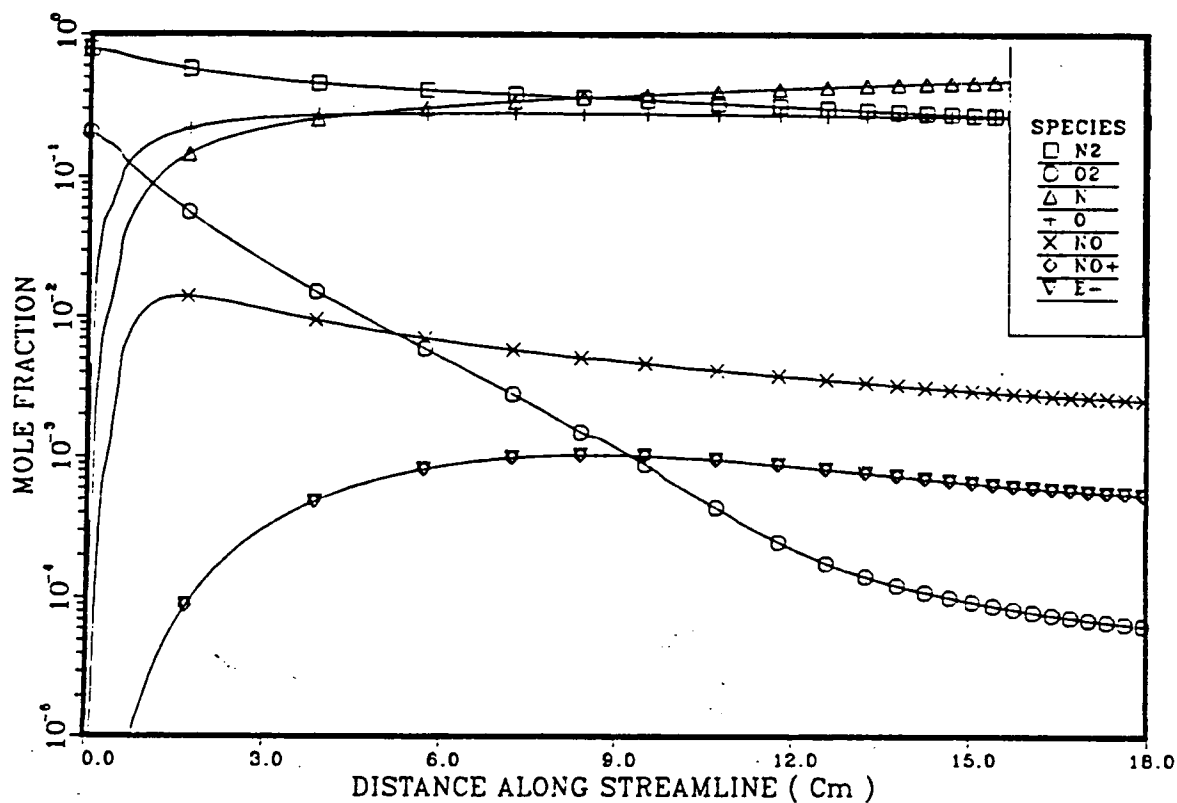
27(a)



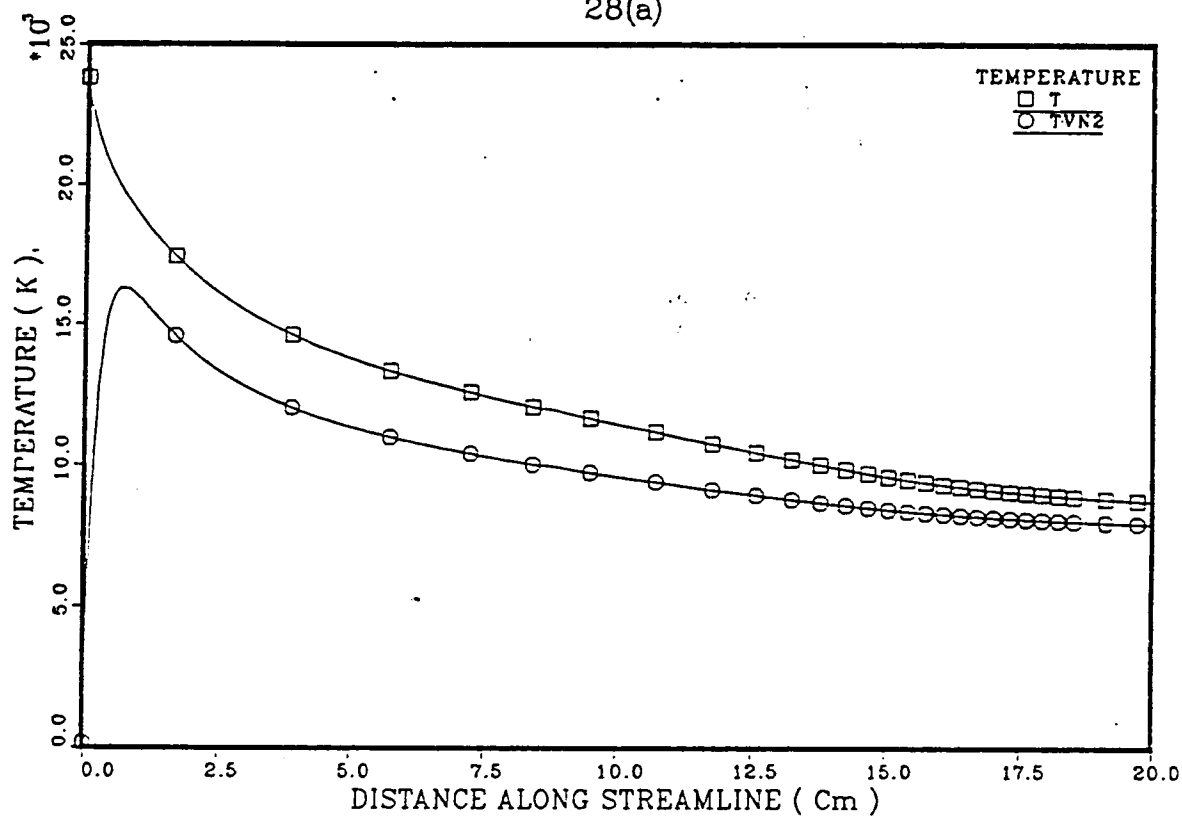
27(b)

FIGURES 27(a),27(b).CVDV MODEL AT V=7.7 Km/s, RR1

204 191

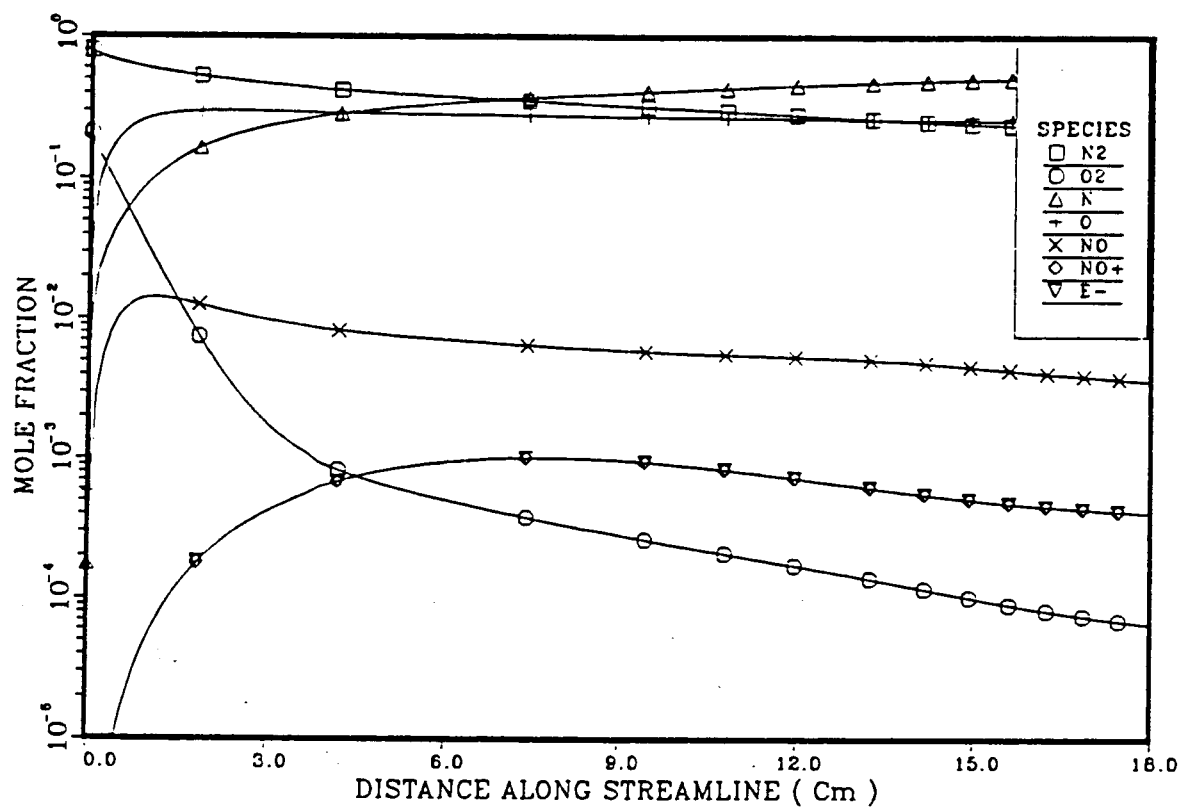


28(a)

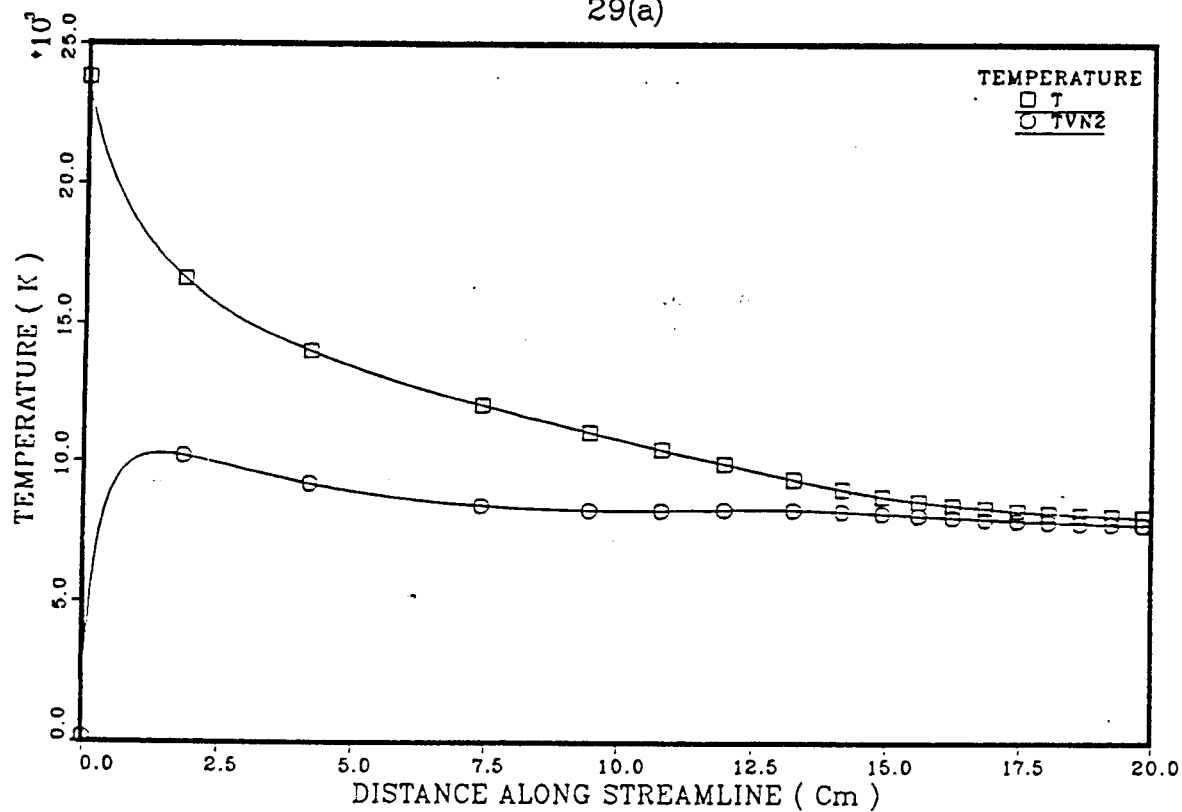


28(b)

FIGURES 28(a),28(b).CVDV-P MODEL AT V=7.7 Km/s, RR1

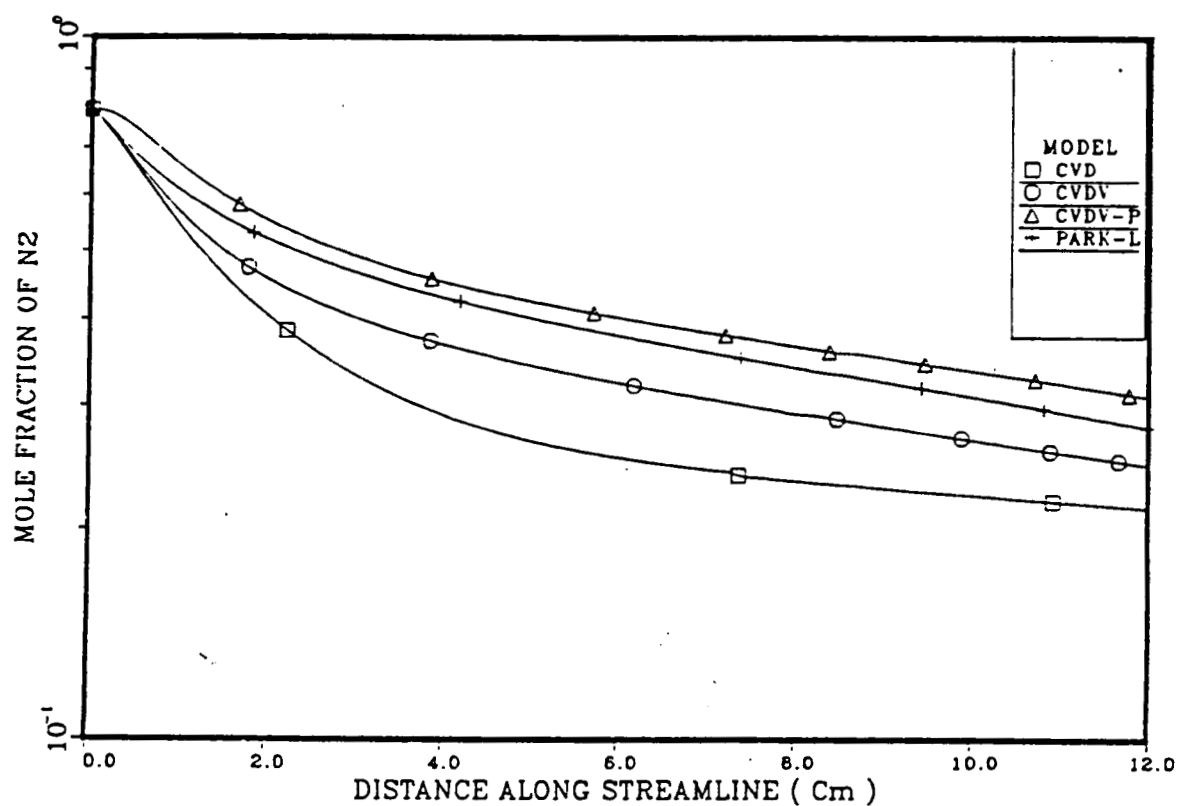


29(a)

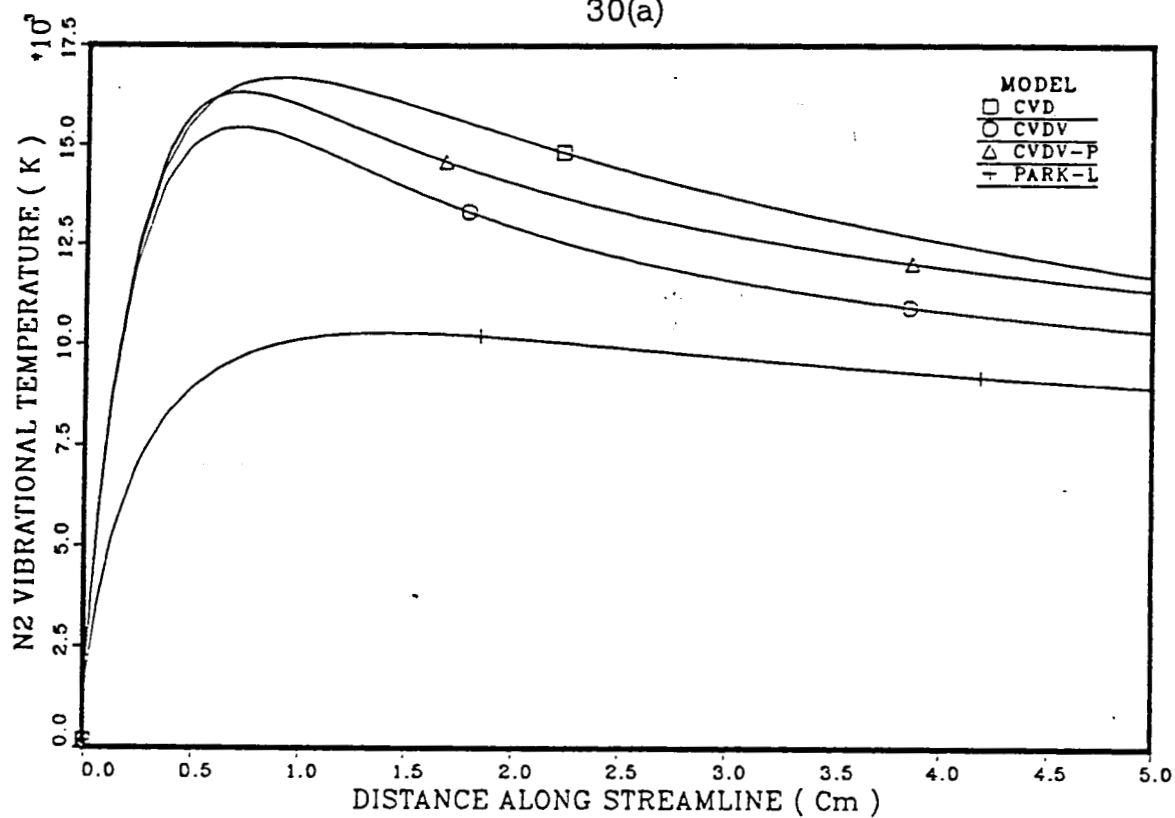


29(b)

FIGURES 29(a),29(b).PARK-L MODEL AT V=7.7 Km/s, RR1

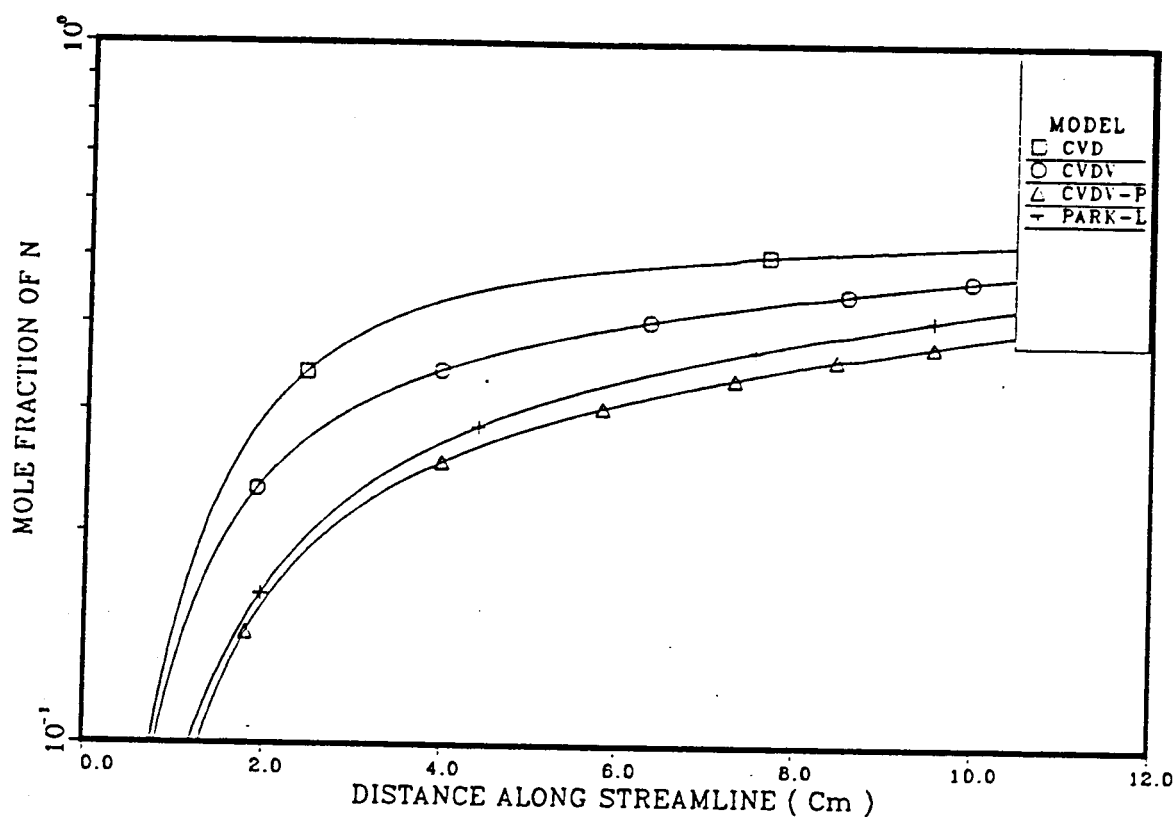


30(a)

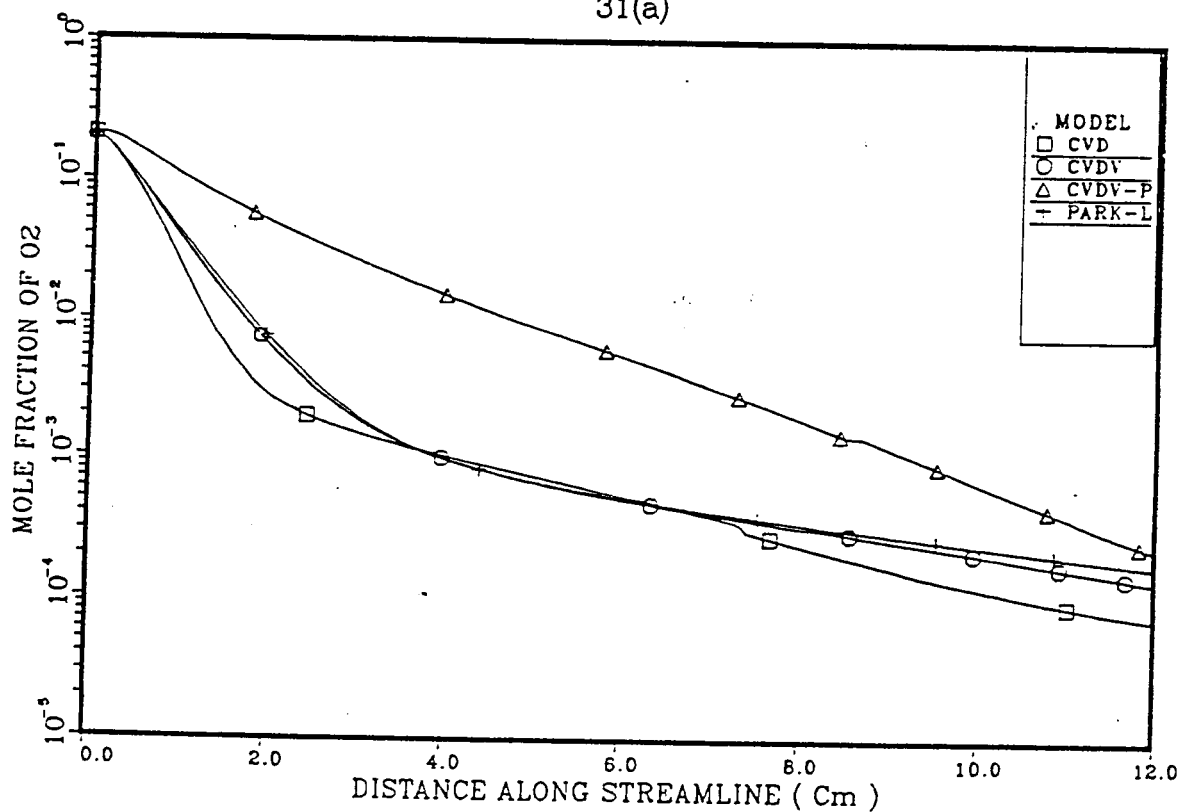


30(b)

FIGURES 30(a),30(b).PROFILES AT V=7.7 Km/s, RR1



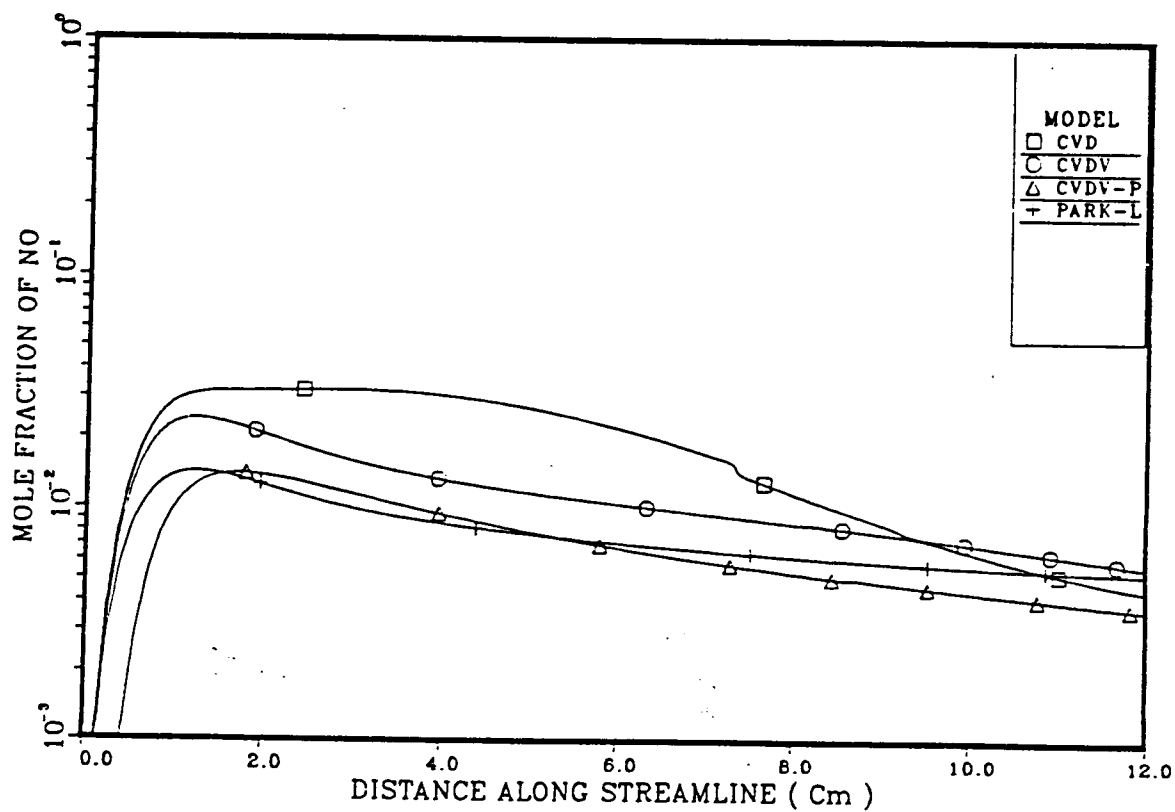
31(a)



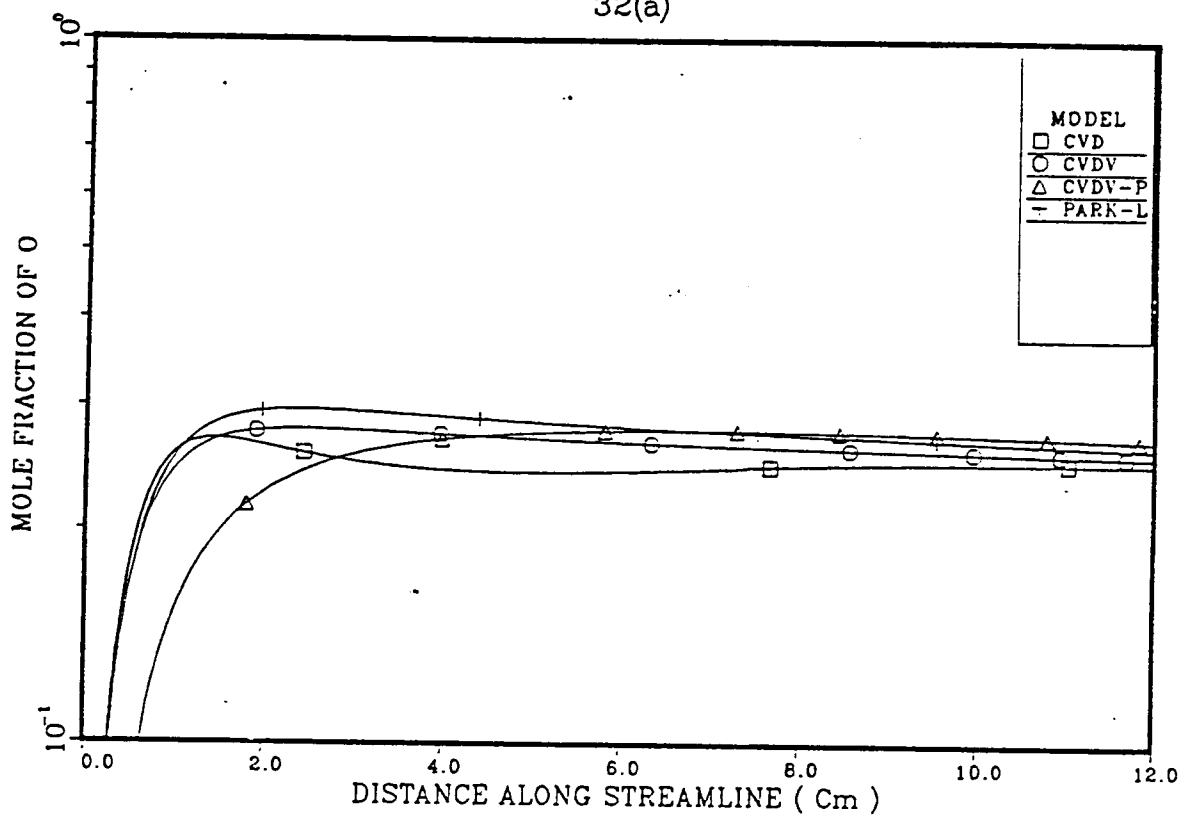
31(b)

FIGURES 31(a),31(b).PROFILES AT V=7.7 Km/s, RR1

205195

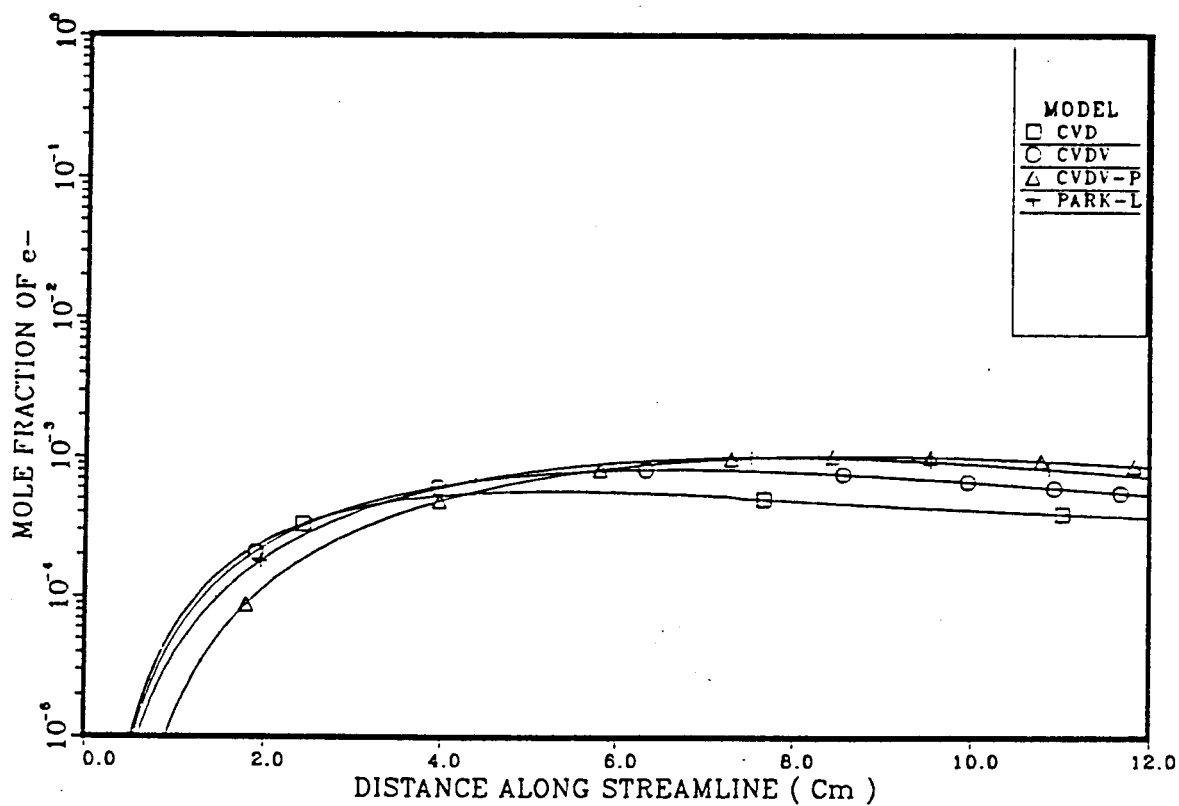


32(a)

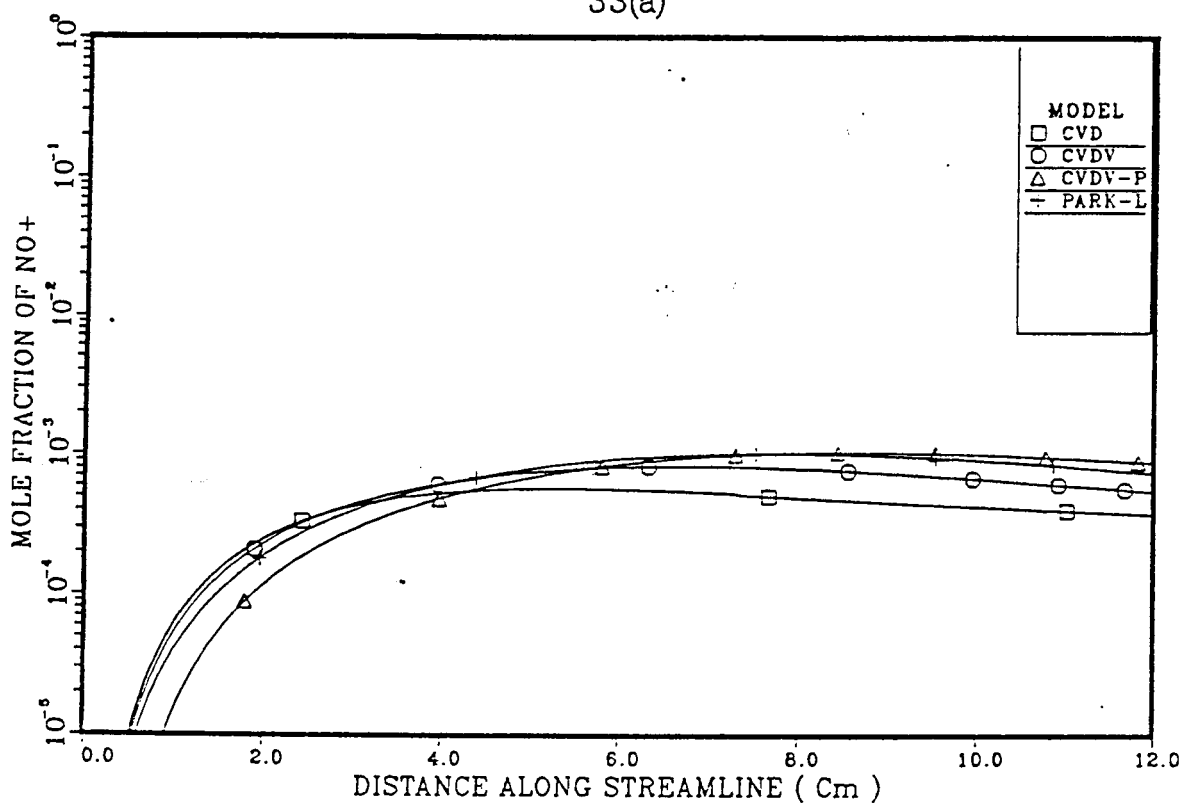


32(b)

FIGURES 32(a),32(b).PROFILES AT V=7.7 Km/s, RR1

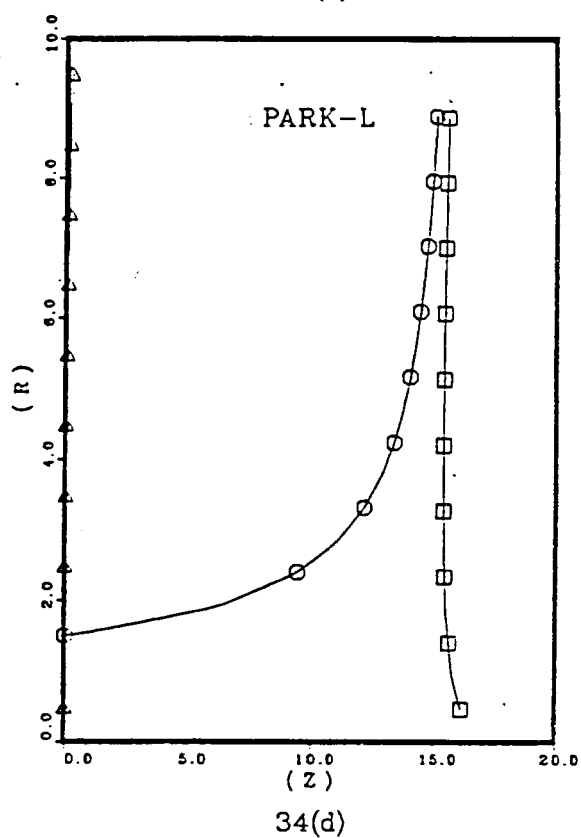
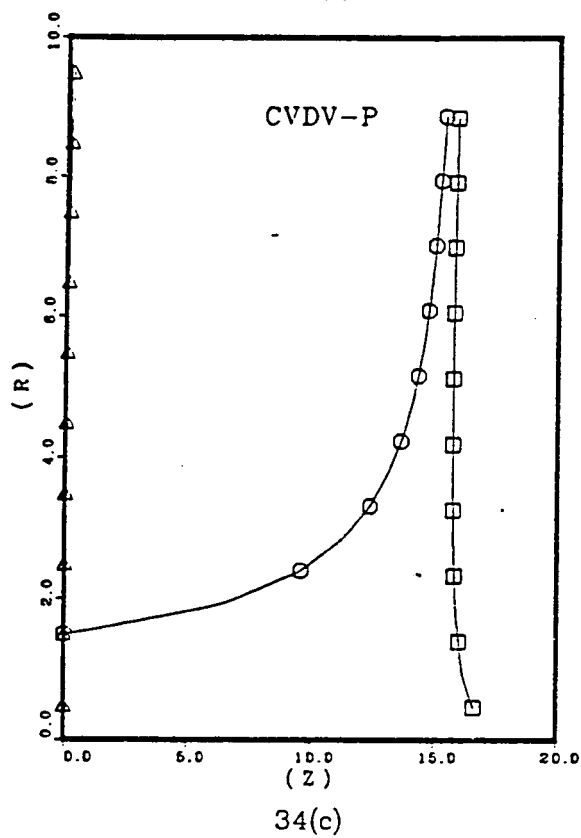
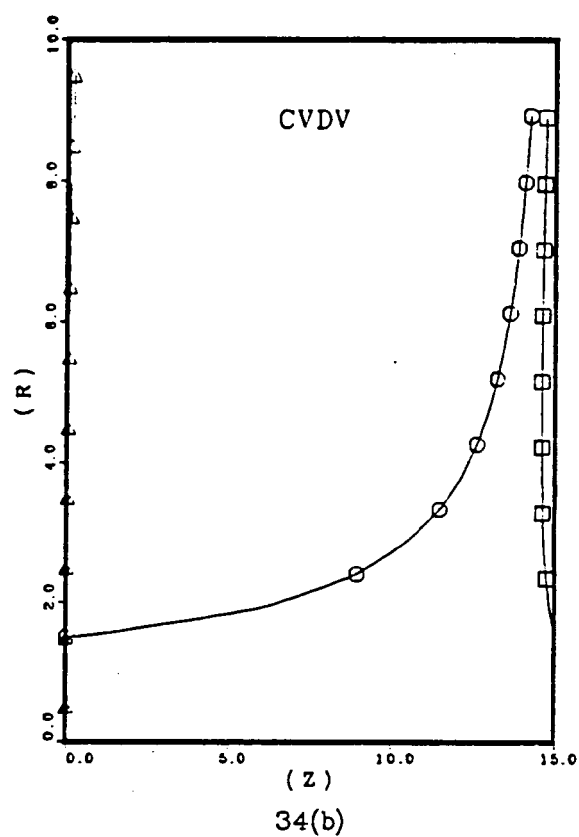
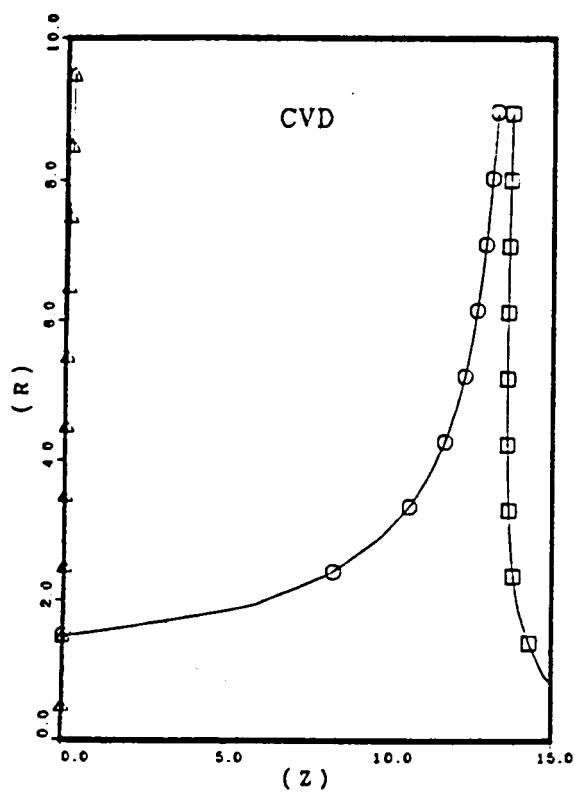


33(a)



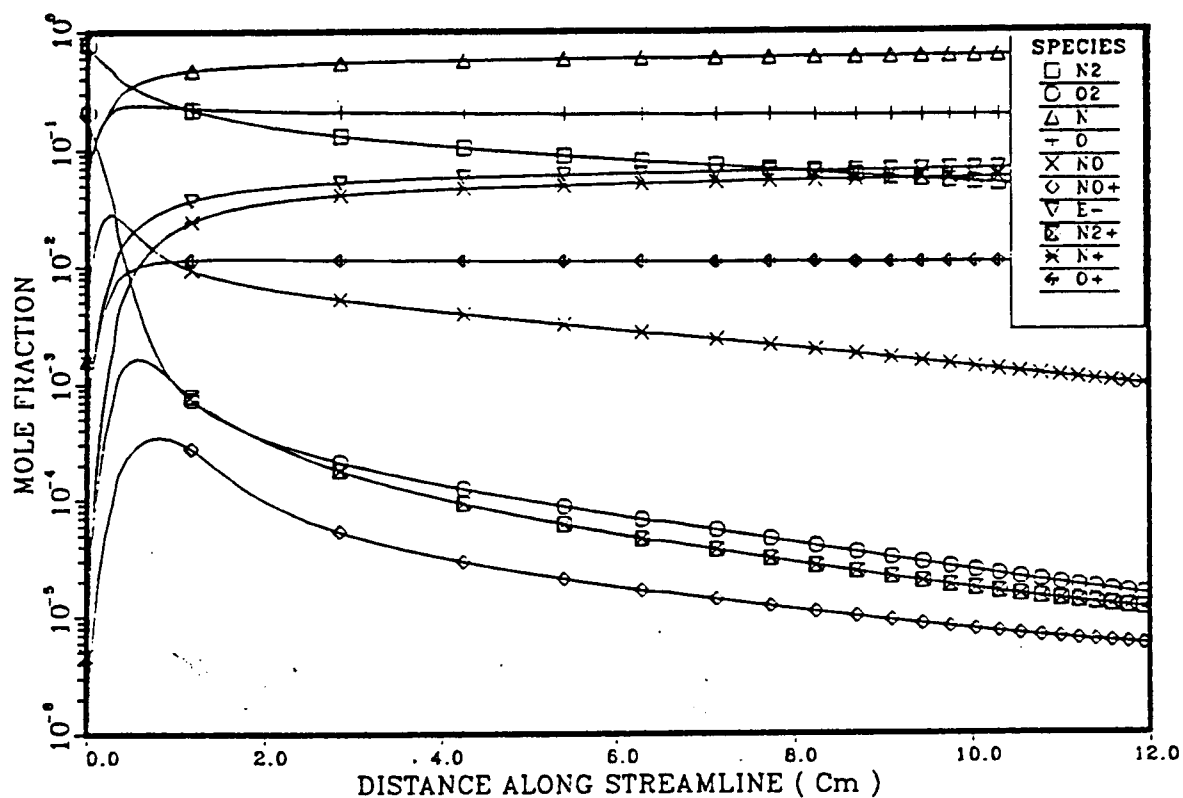
33(b)

FIGURES 33(a),33(b).PROFILES AT $V=7.7$ Km/s, RR1

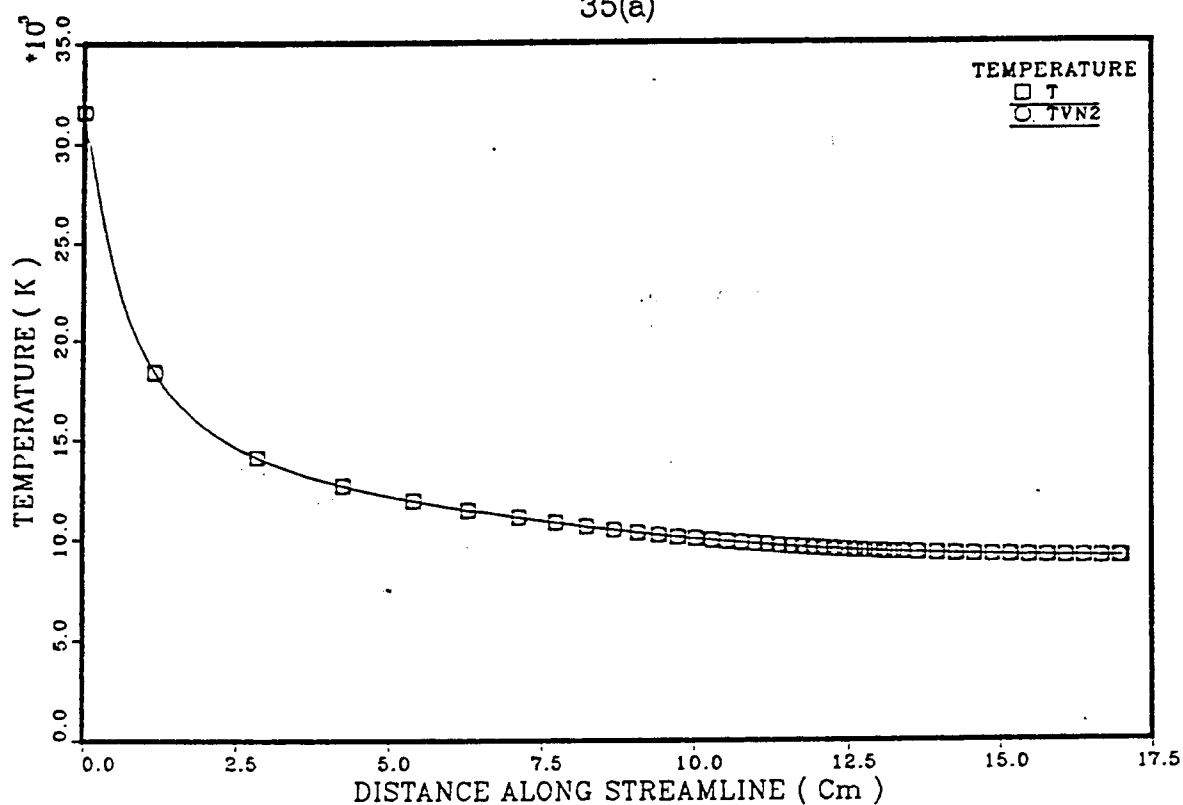


FIGURES 34(a),34(b),34(c),34(d).COORD,V=7.7 Km/s, RR1

209 199



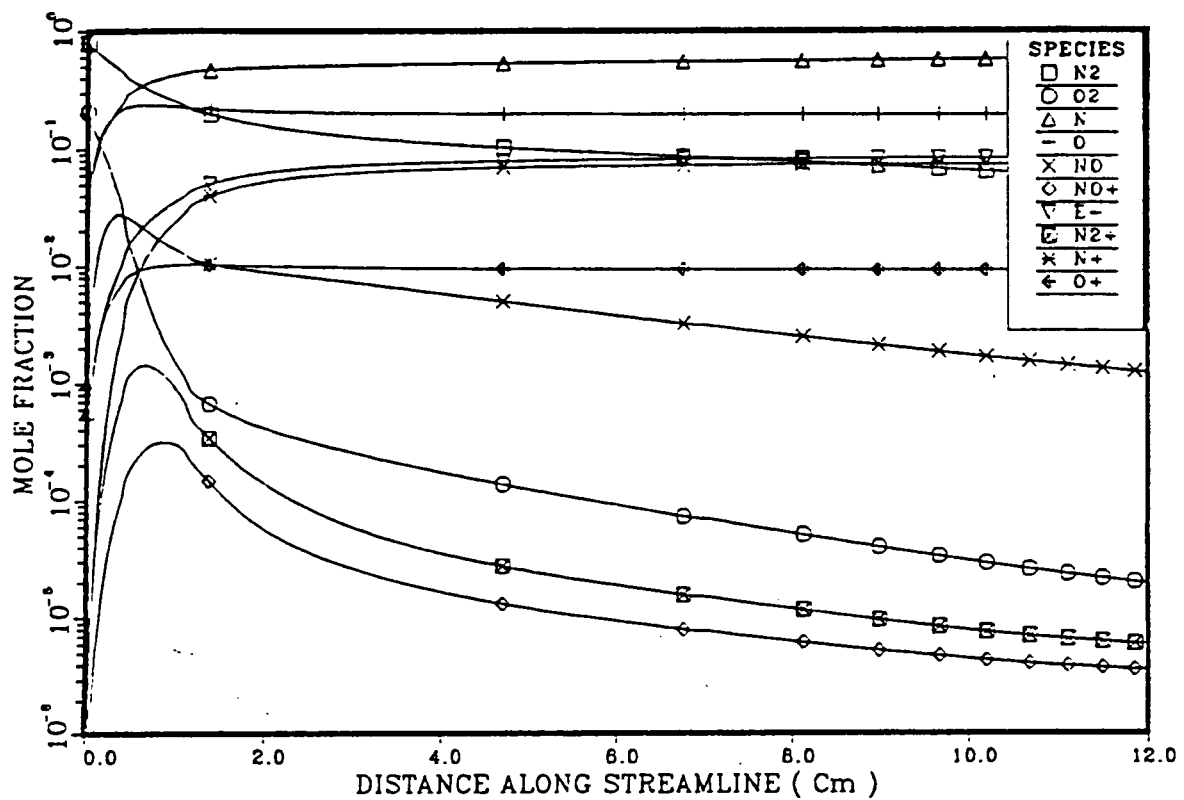
35(a)



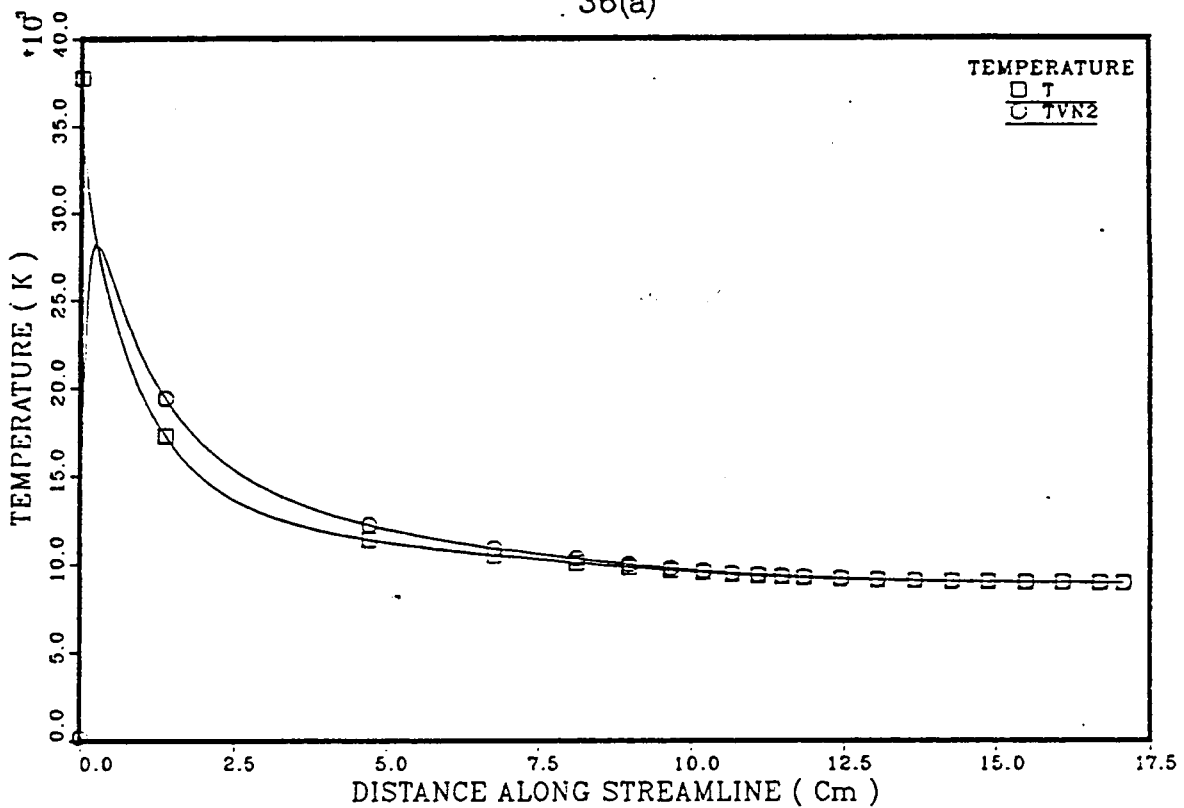
35(b)

FIGURES 35(a),35(b).VEQ MODEL AT V=10 Km/s, RR2

209 199

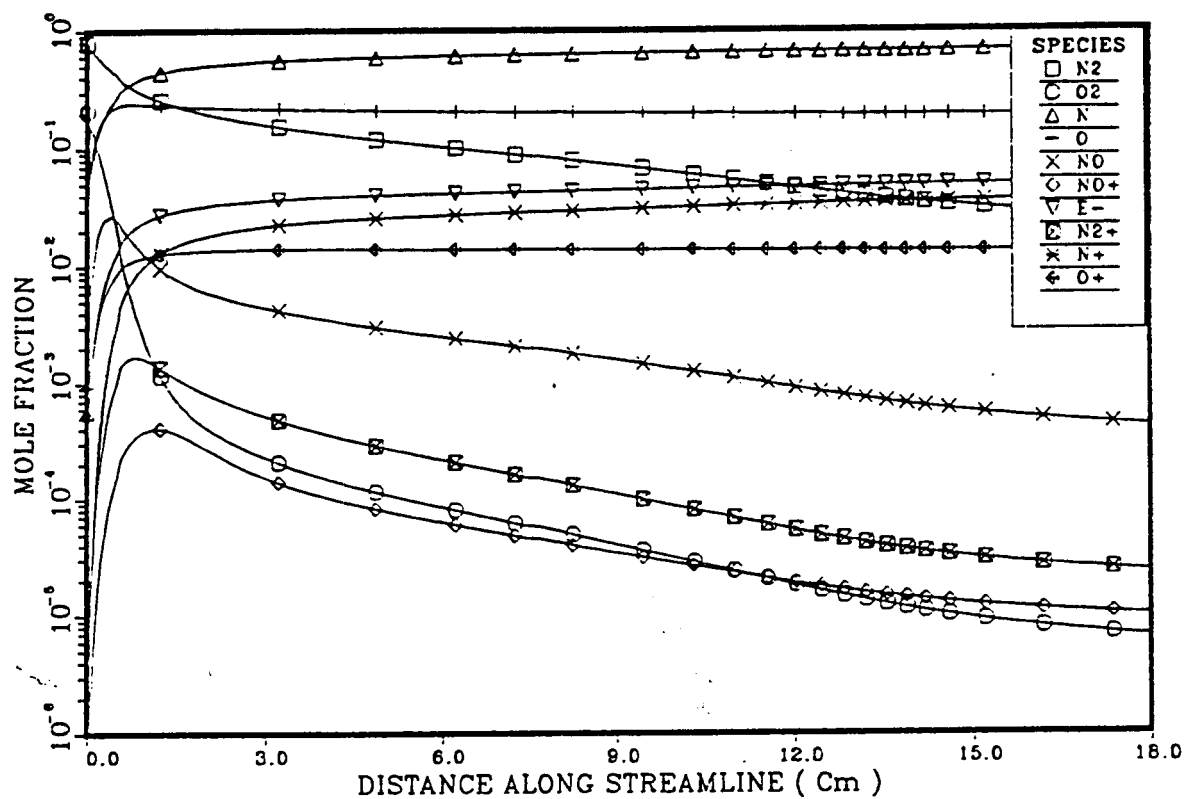


36(a)

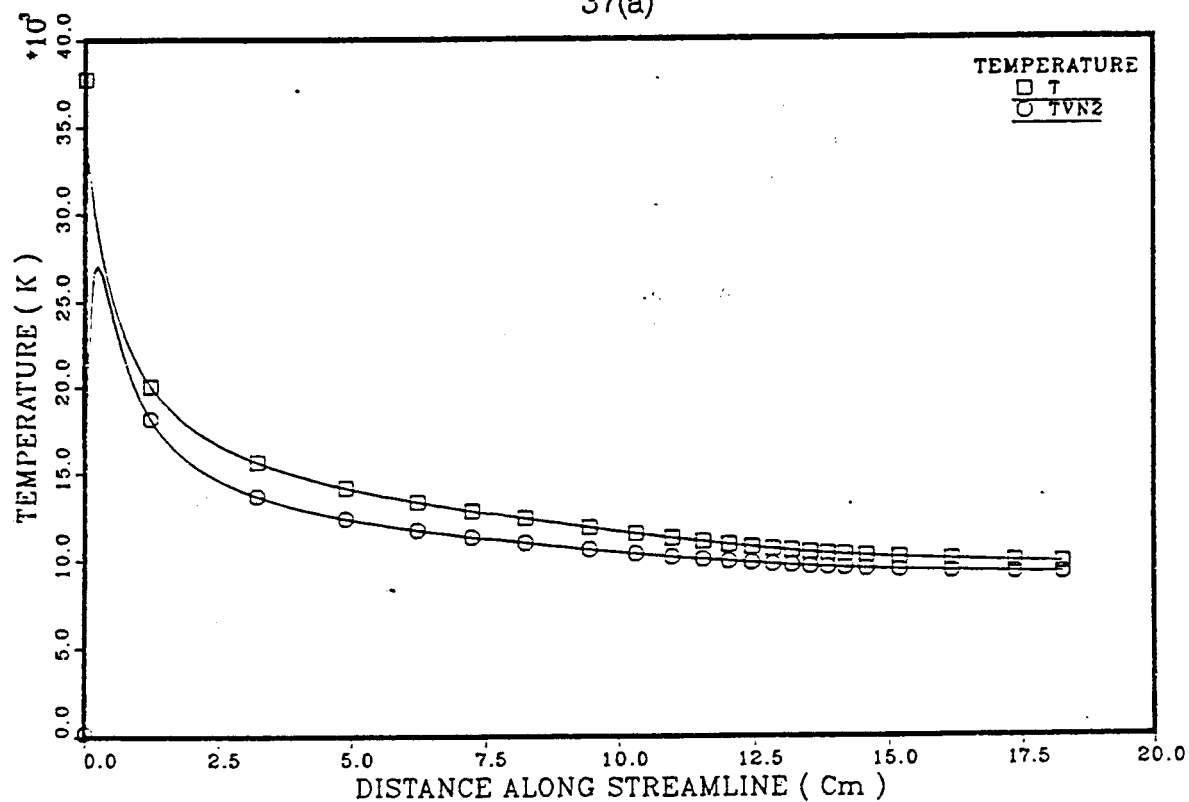


36(b)

FIGURES 36(a),36(b).CVD MODEL AT V=10 Km/s, RR2

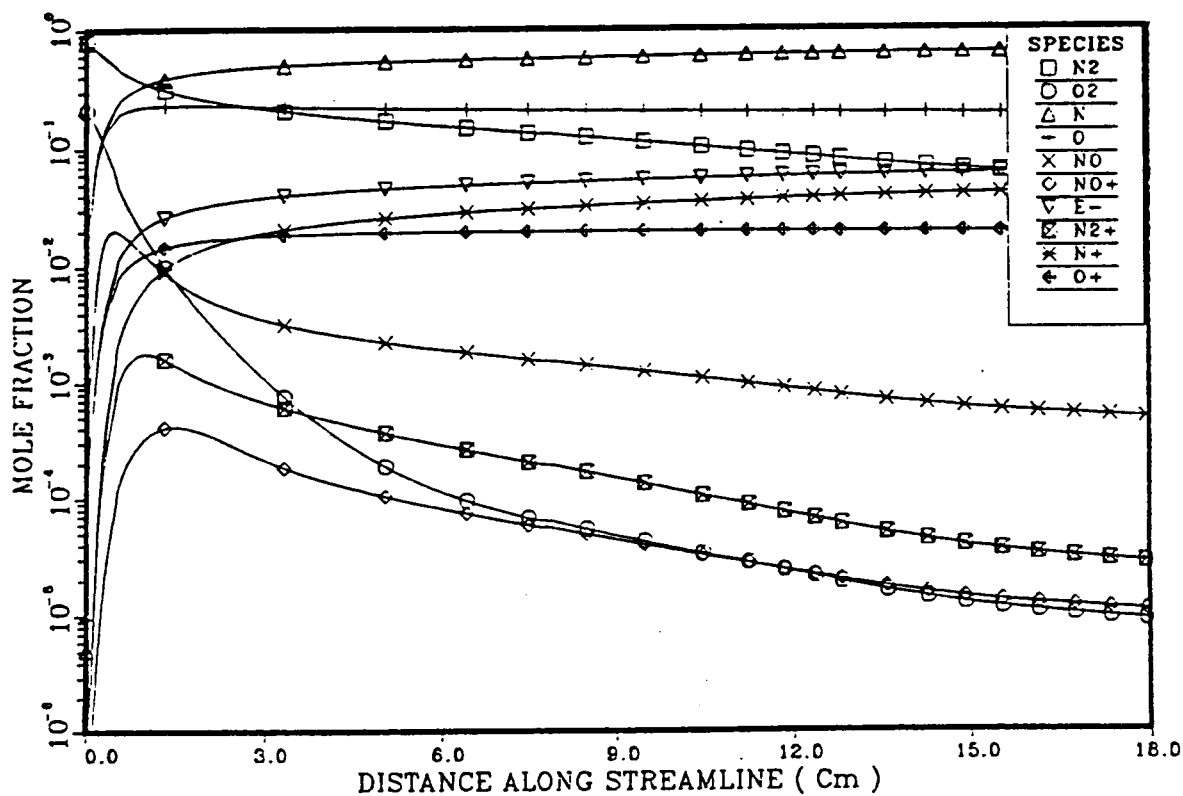


37(a)

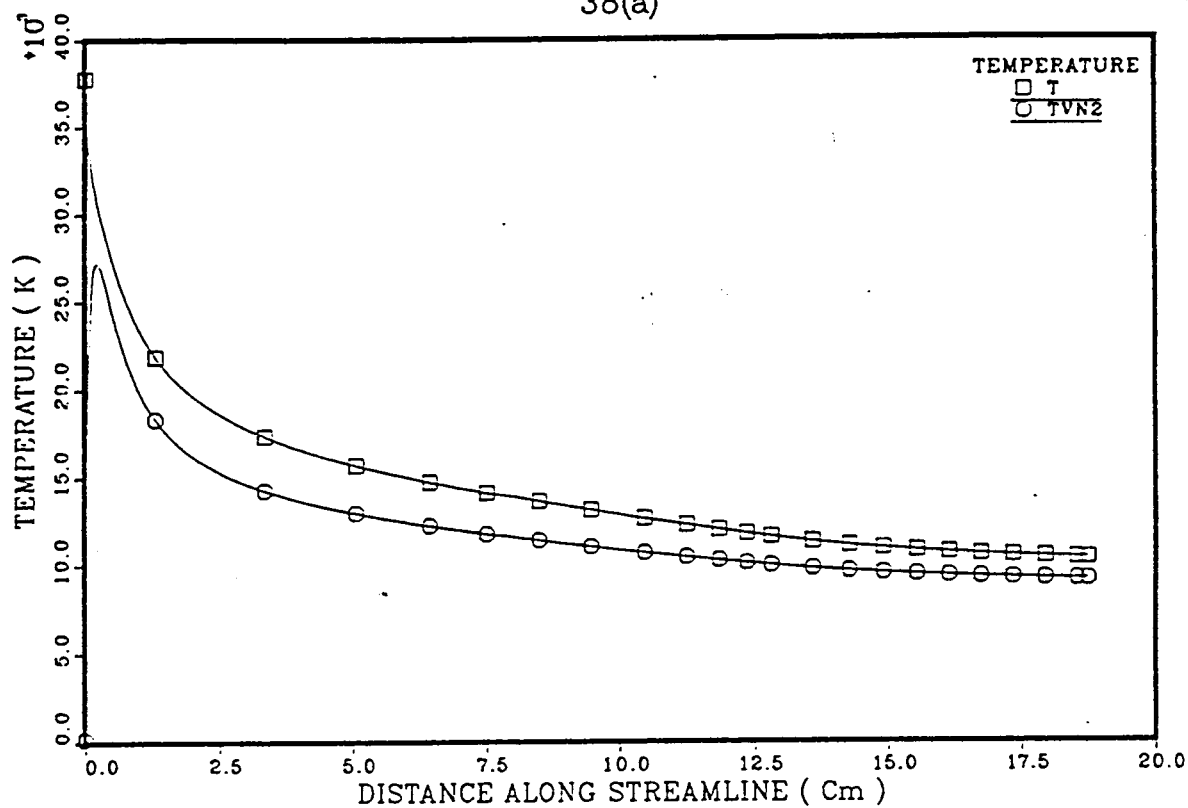


37(b)

FIGURES 37(a),37(b).CVDV MODEL AT V=10 Km/s, RR2

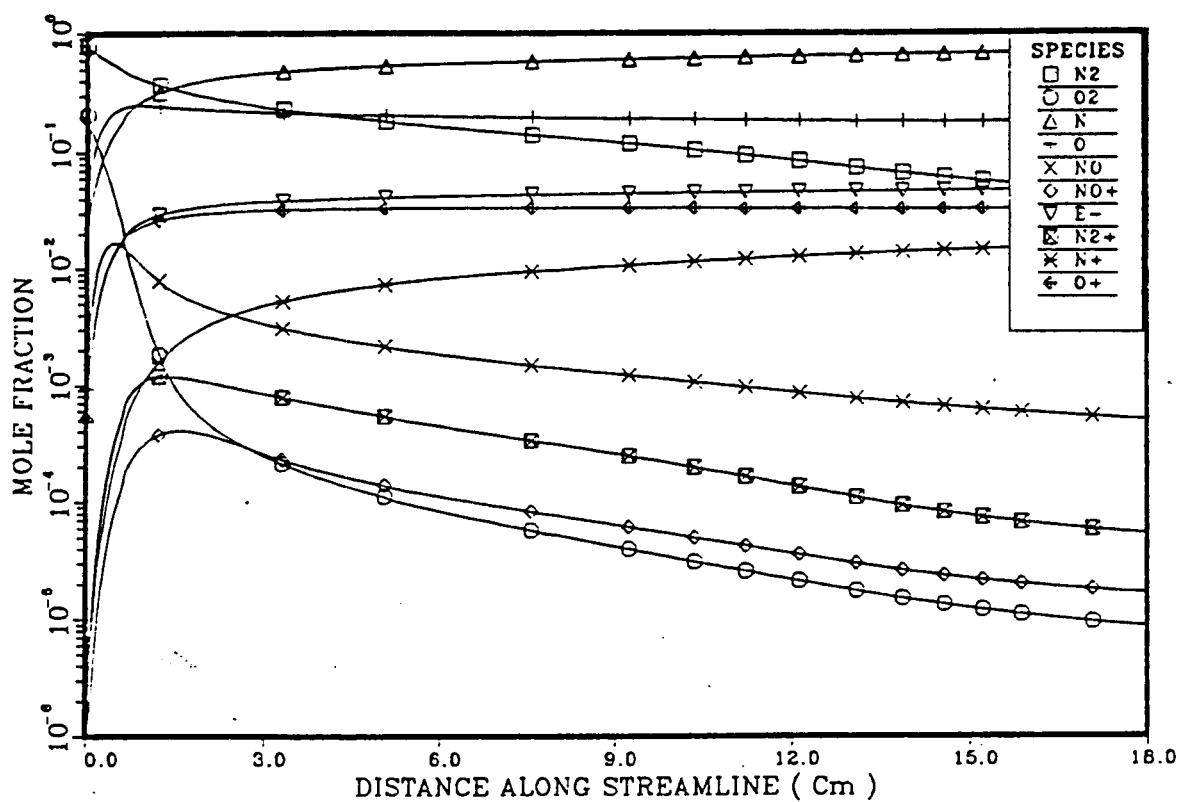


38(a)

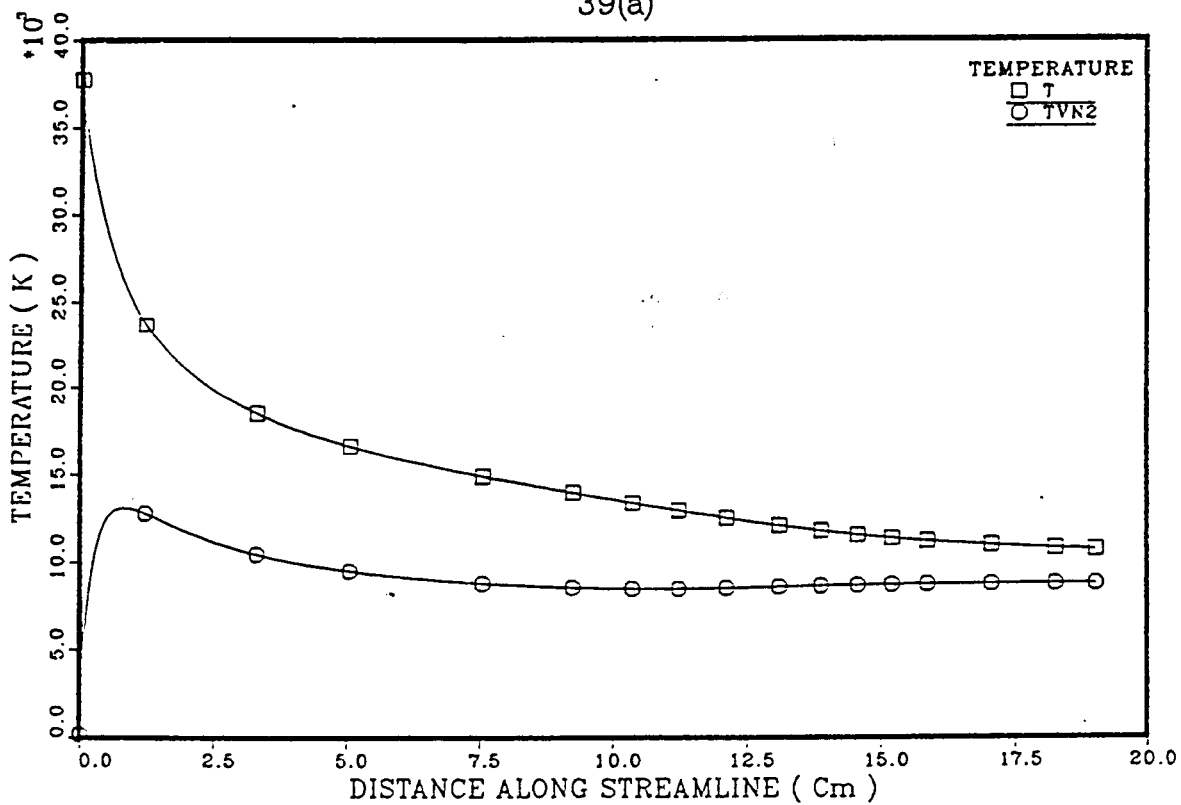


38(b)

FIGURES 38(a),38(b).CVDV-P MODEL AT V=10 Km/s, RR2

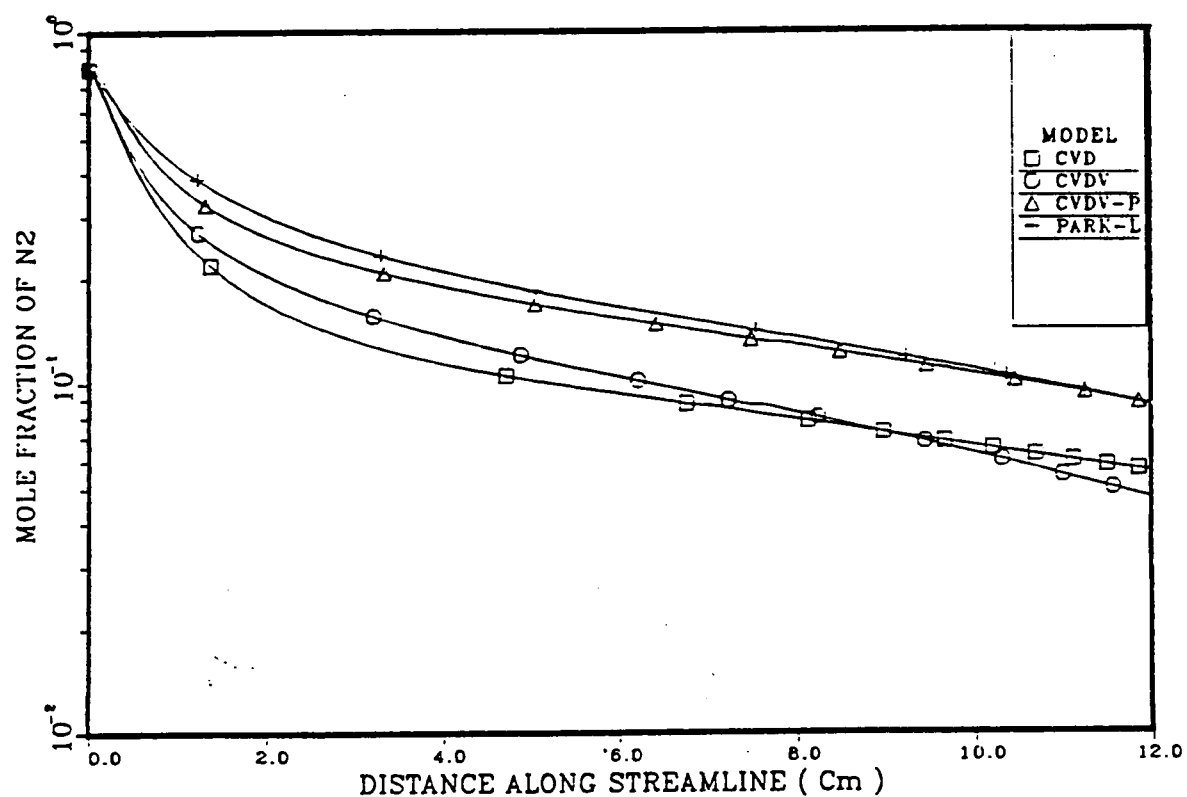


39(a)

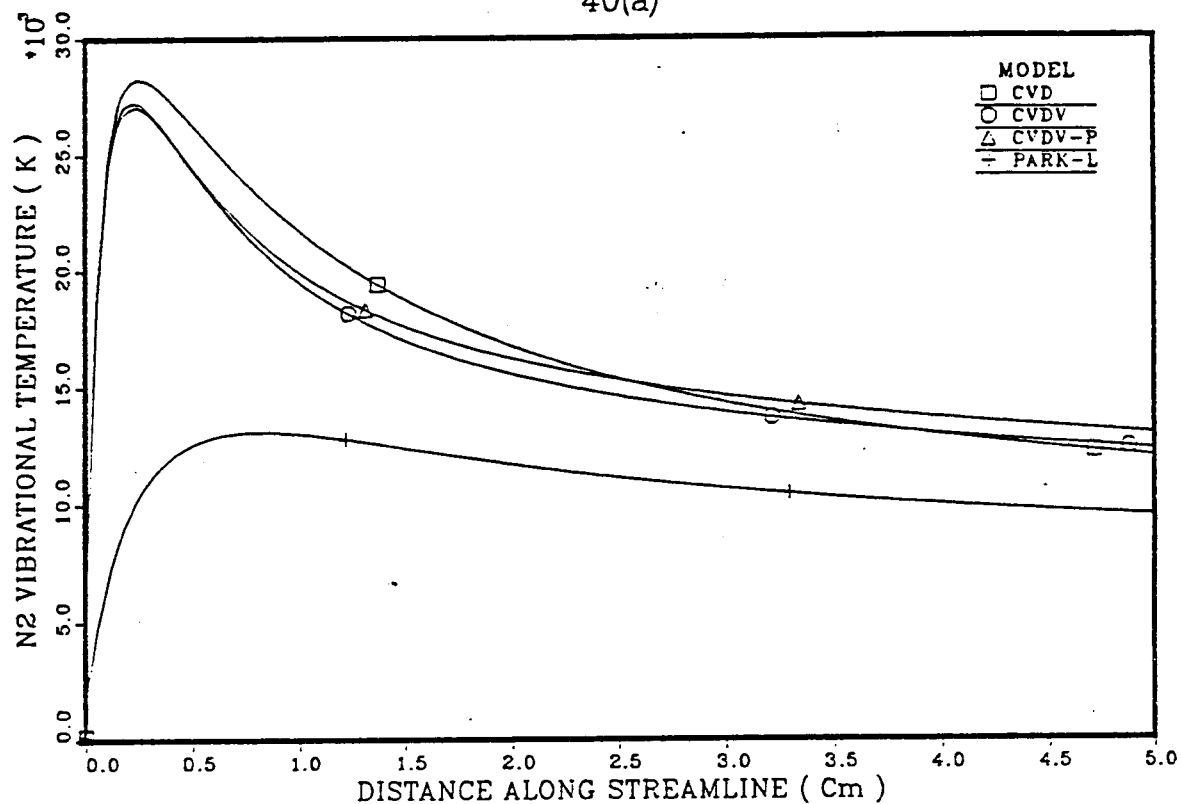


39(b)

FIGURES 39(a),39(b).PARK-L MODEL AT V=10 Km/s, RR2

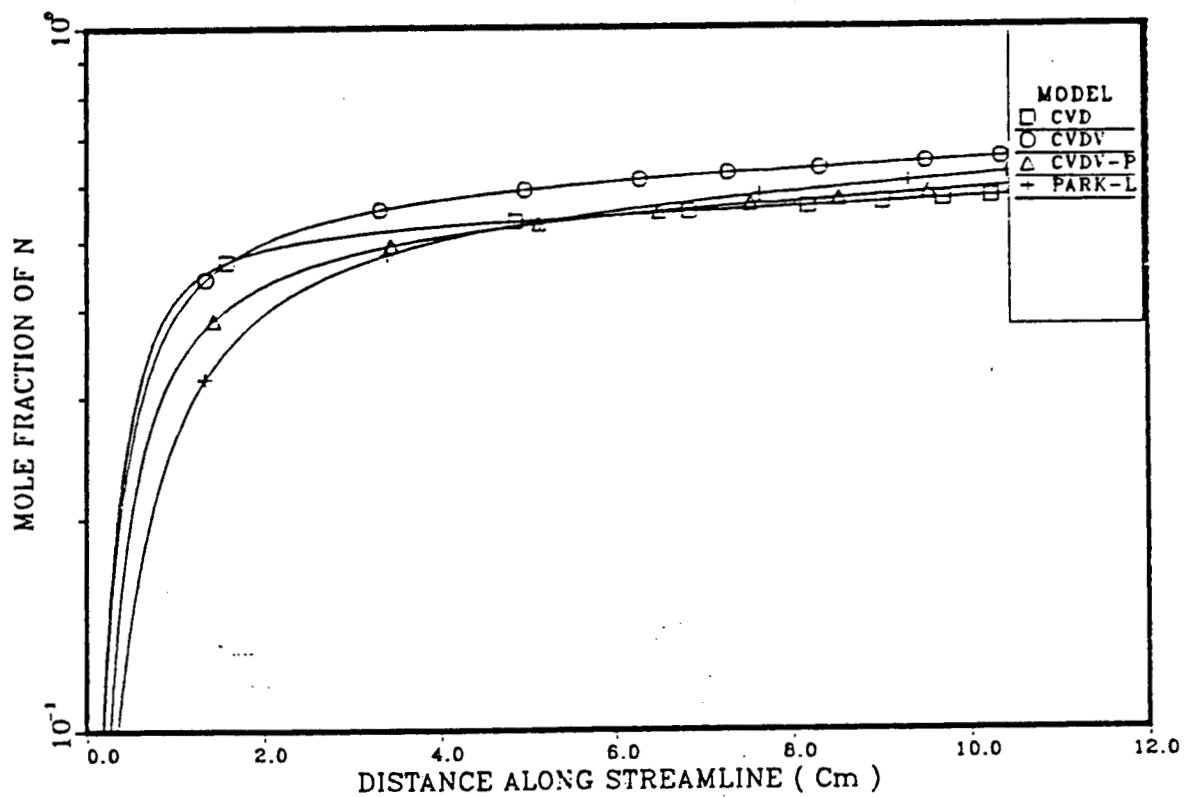


40(a)

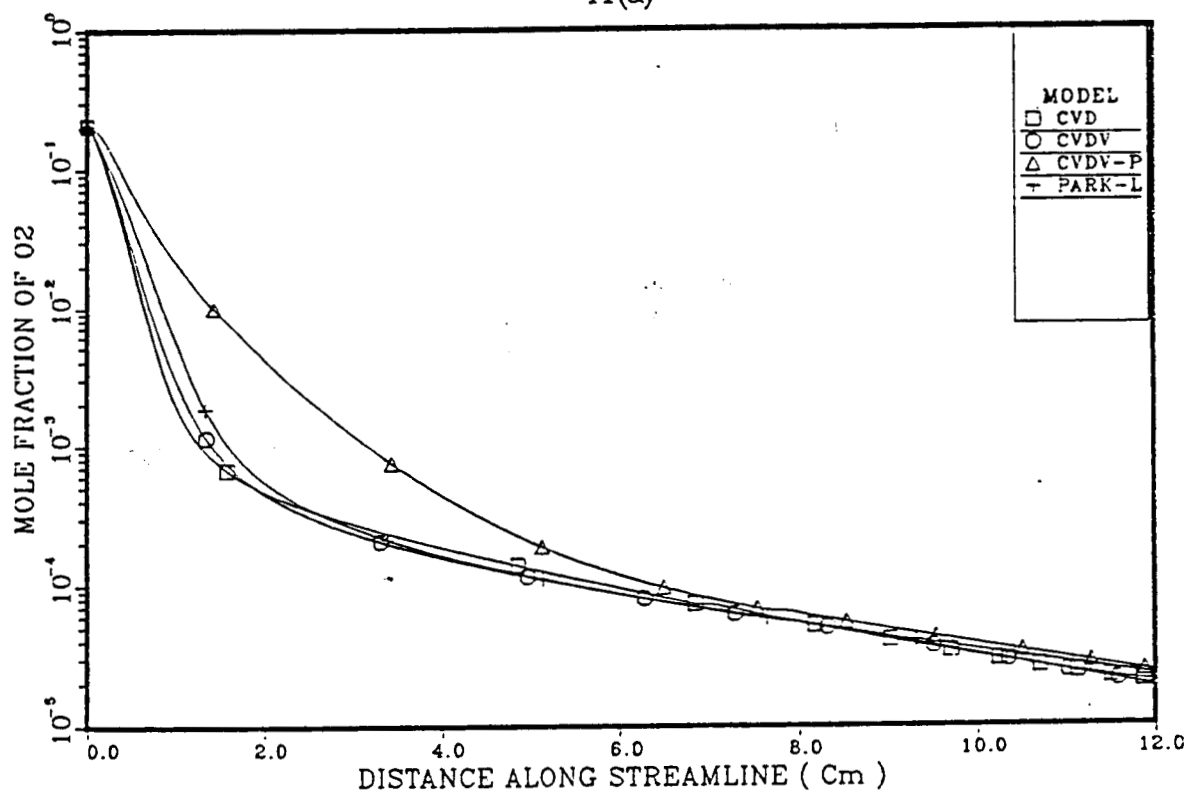


40(b)

FIGURES 40(a),40(b).PROFILES AT V=10 Km/s, RR2

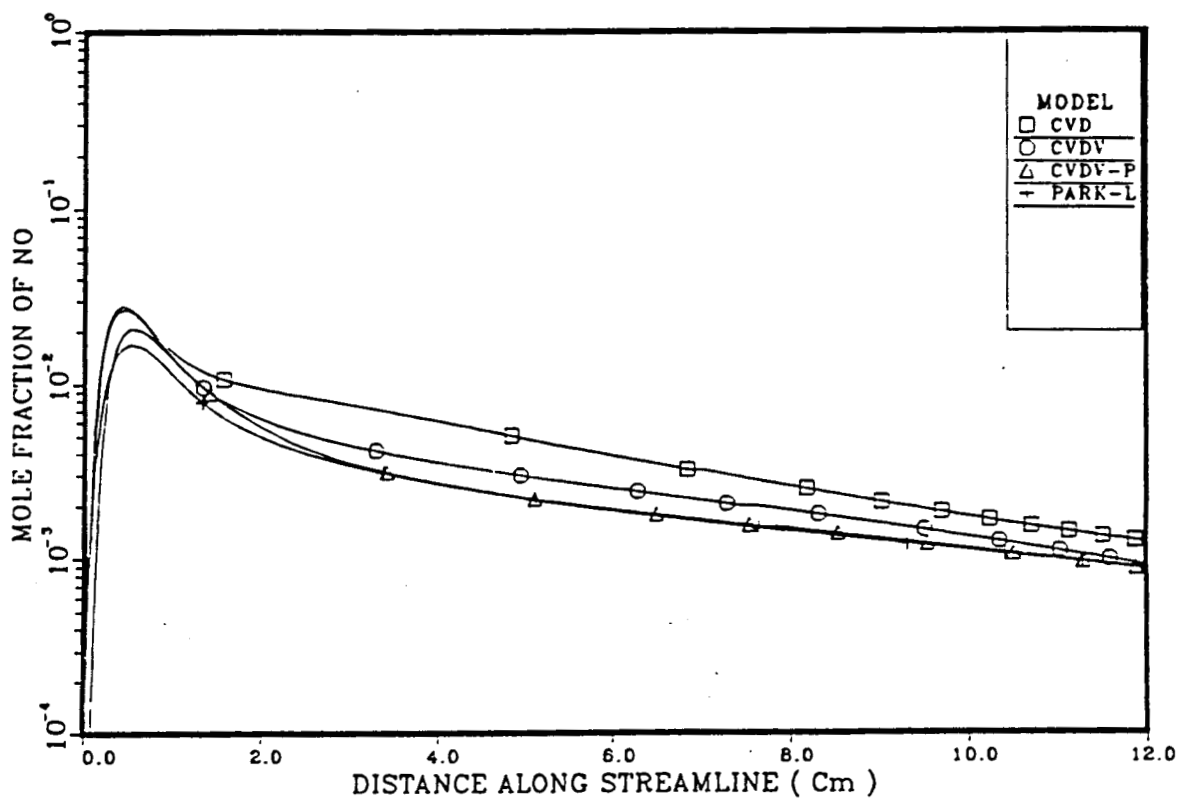


41(a)

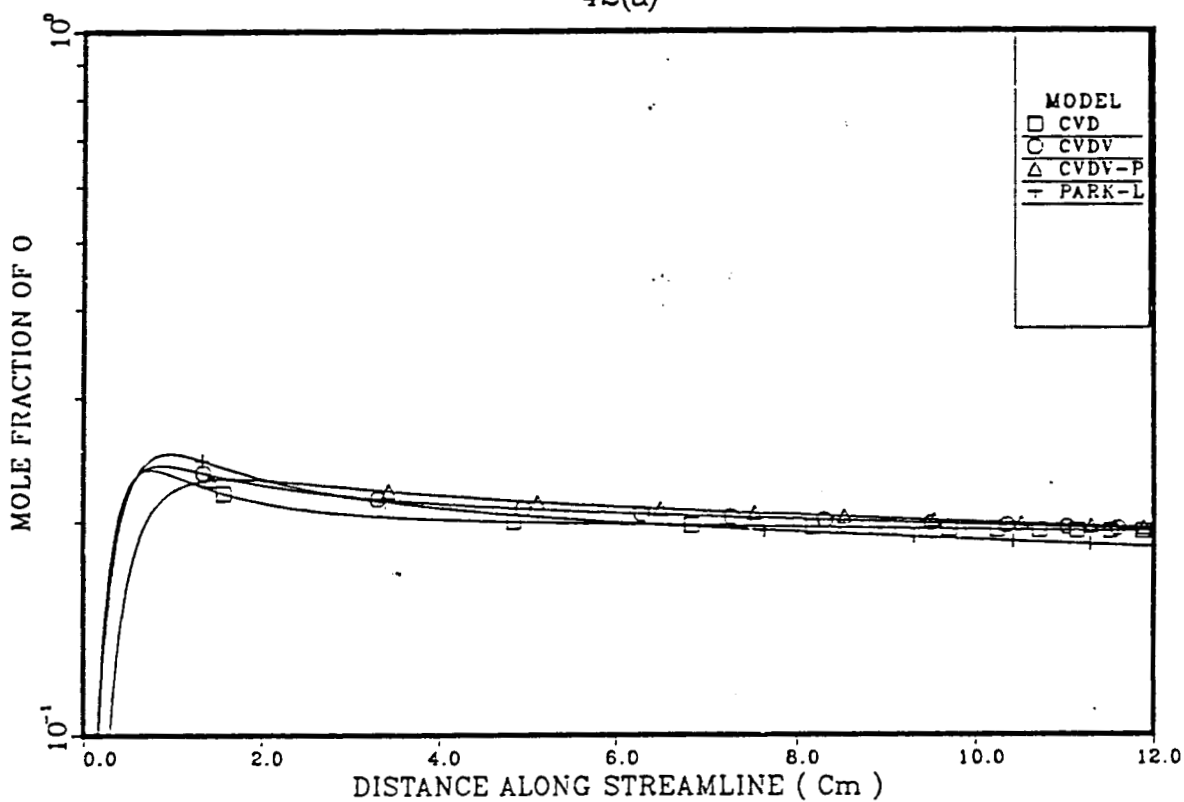


41(b)

FIGURES 41(a),41(b).PROFILES AT V=10 Km/s, RR2

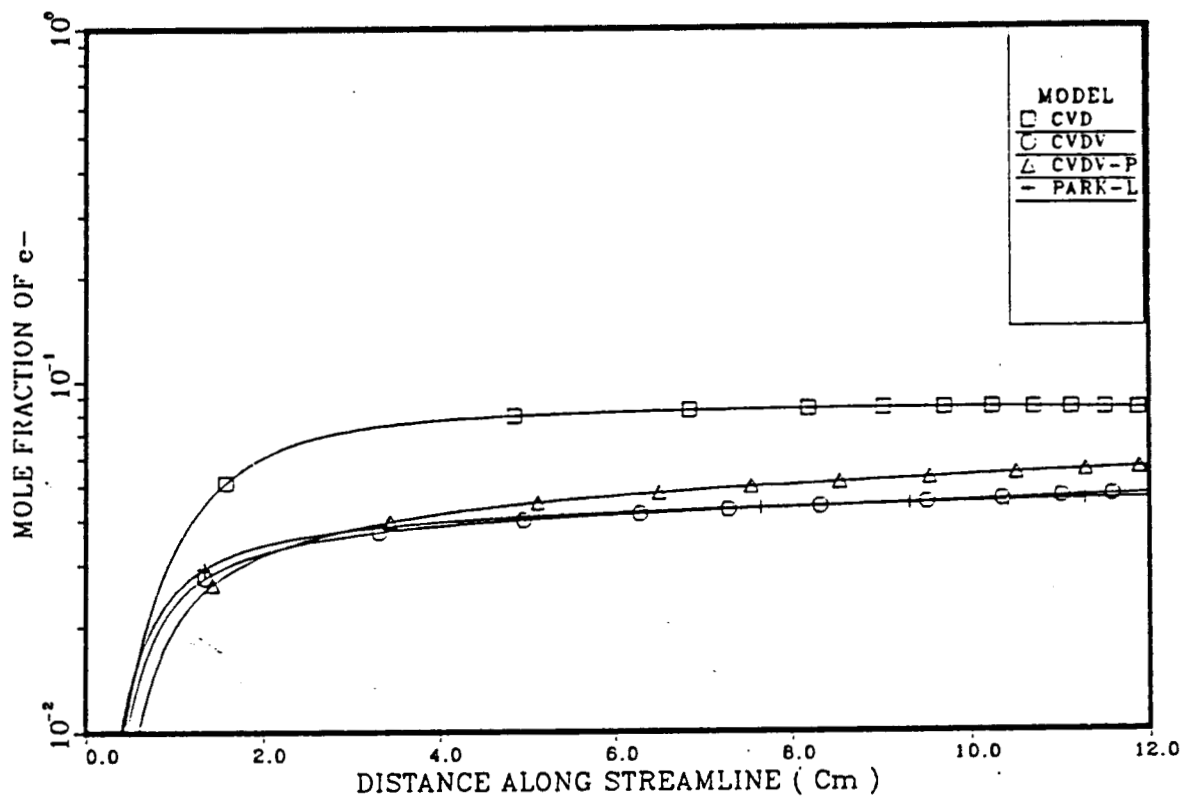


42(a)

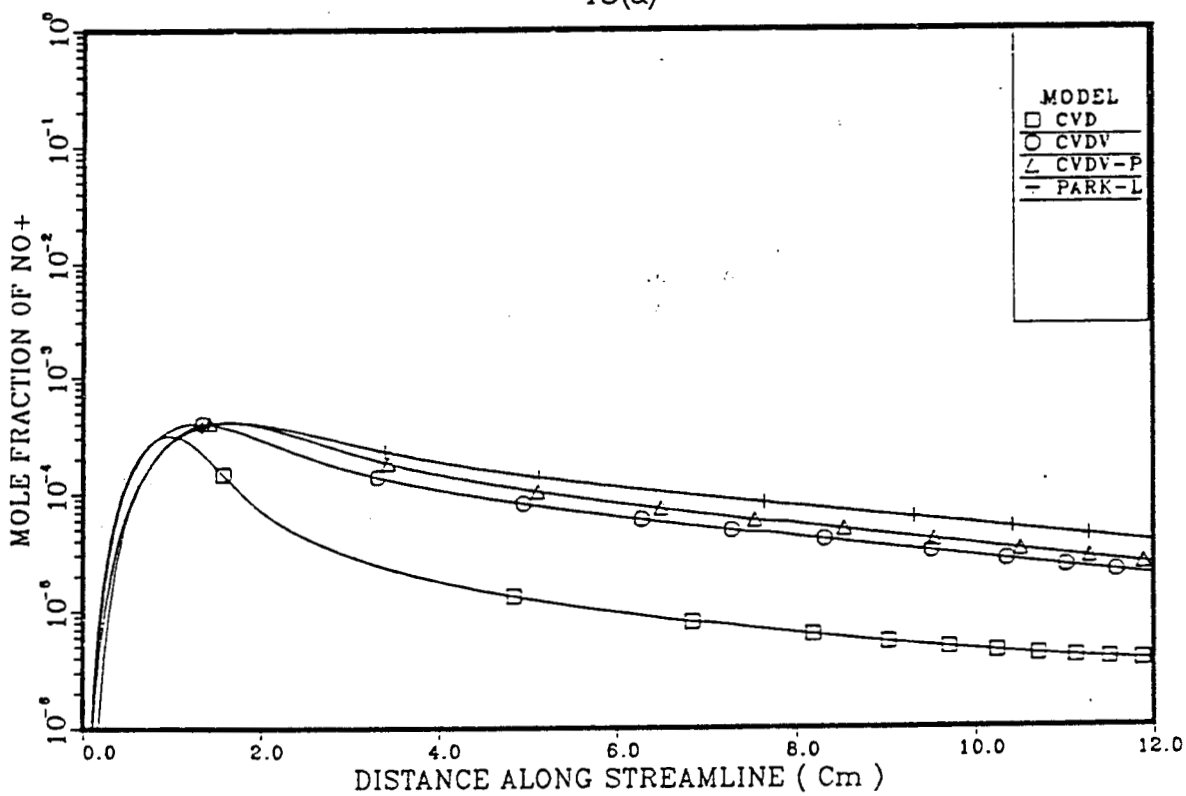


42(b)

FIGURES 42(a),42(b).PROFILES AT V=10 Km/s, RR2

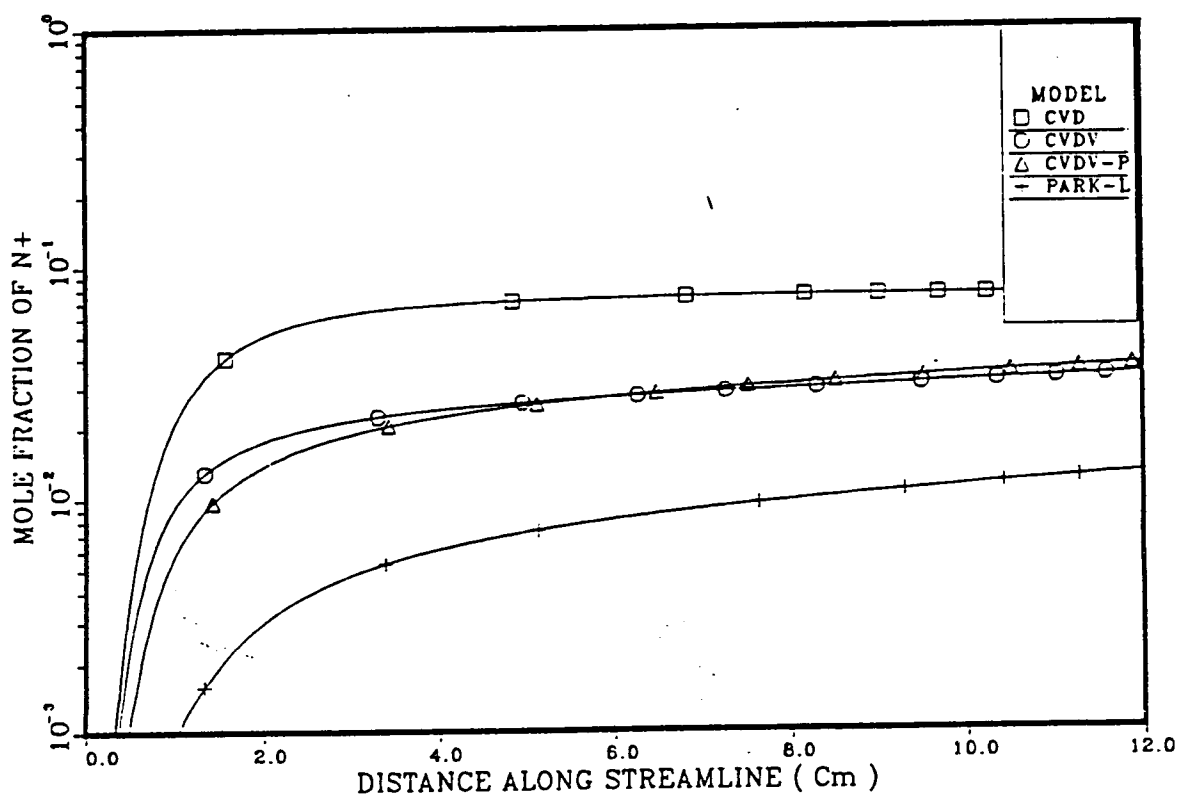


43(a)

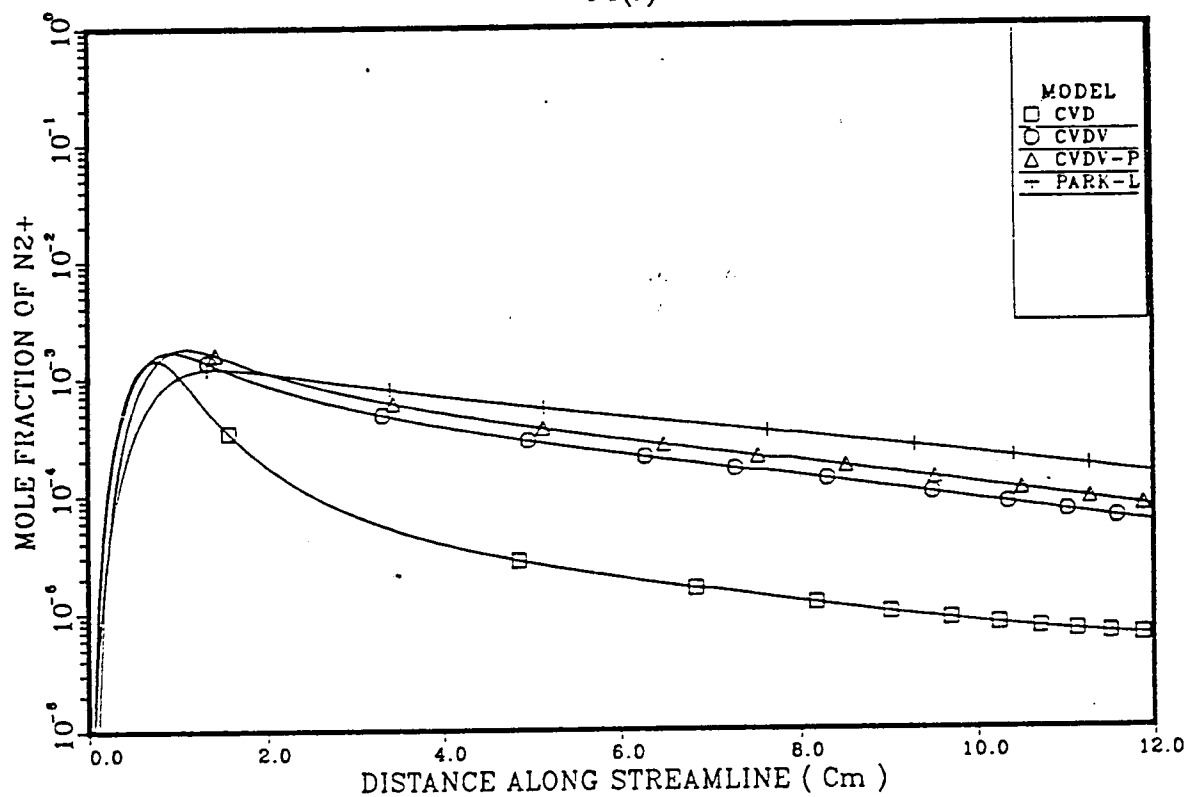


43(b)

FIGURES 43(a), 43(b). PROFILES AT $V=10$ Km/s, RR2



44(a)



44(b)

FIGURES 44(a),44(b).PROFILES AT $V=10$ Km/s, RR2

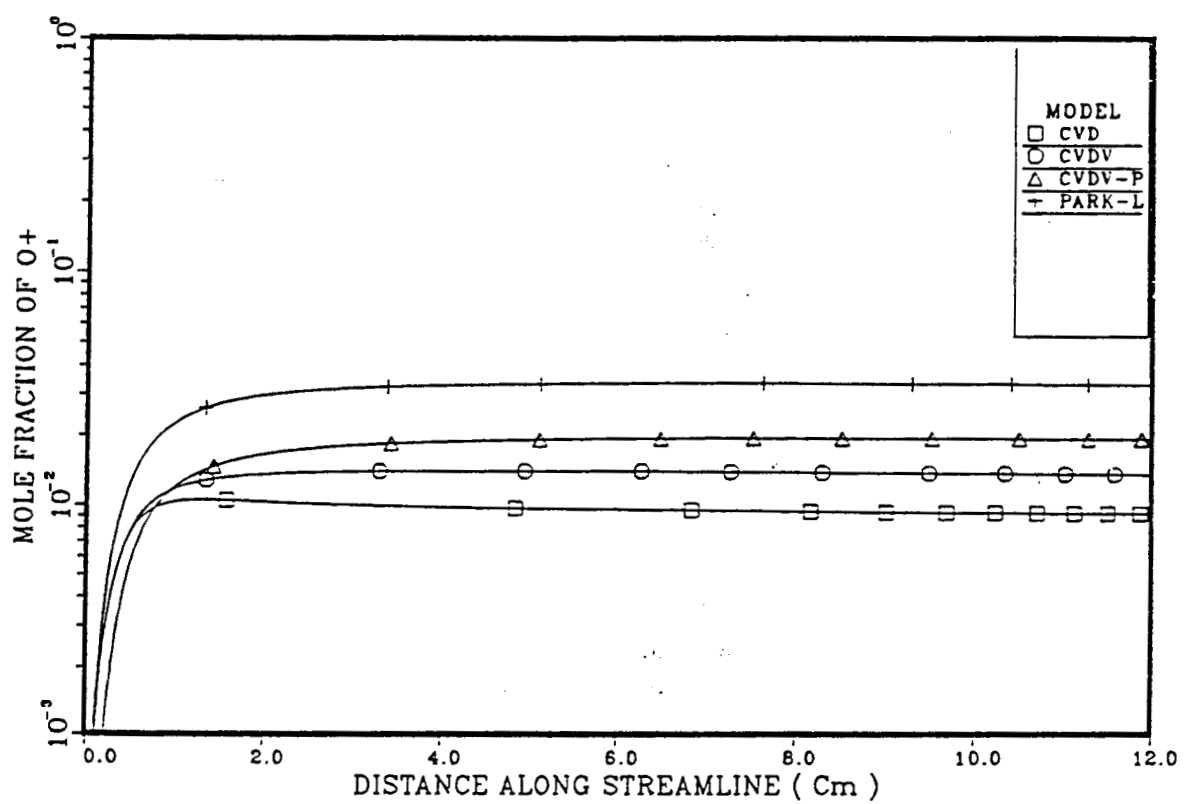
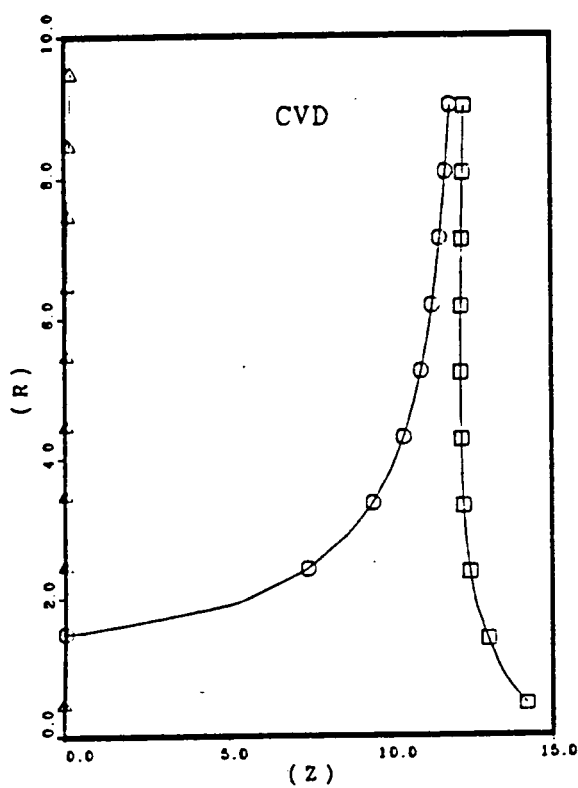
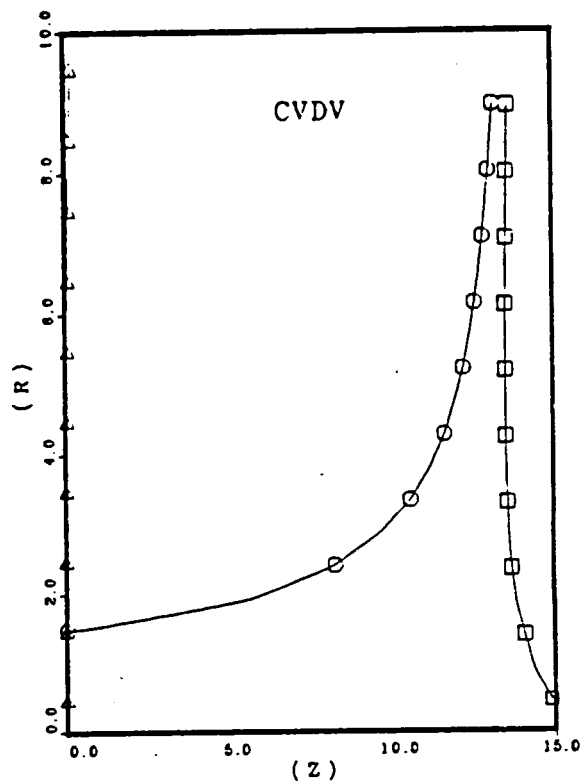


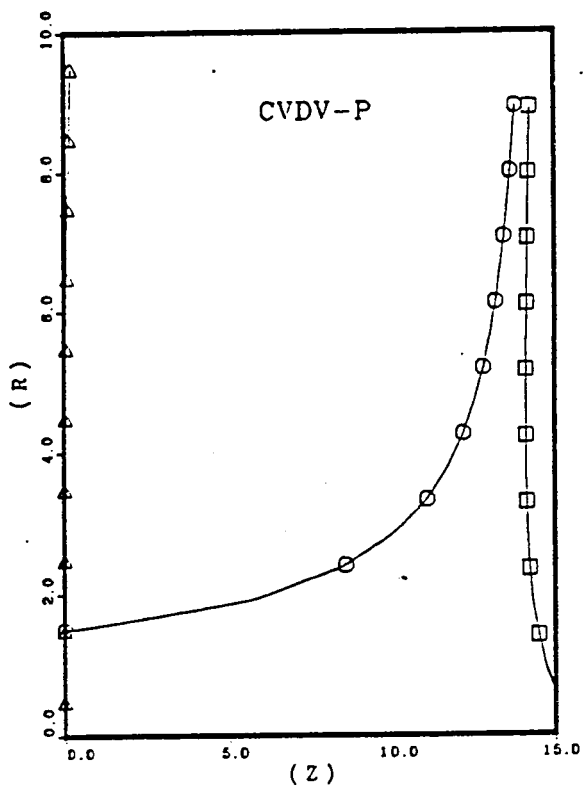
FIGURE 45.PROFILE AT V=10 Km/s, RR2



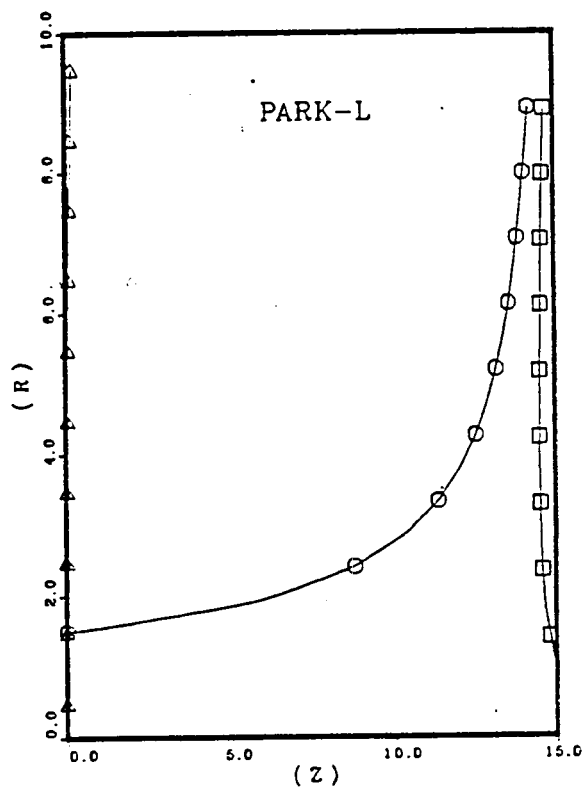
46(a)



46(b)

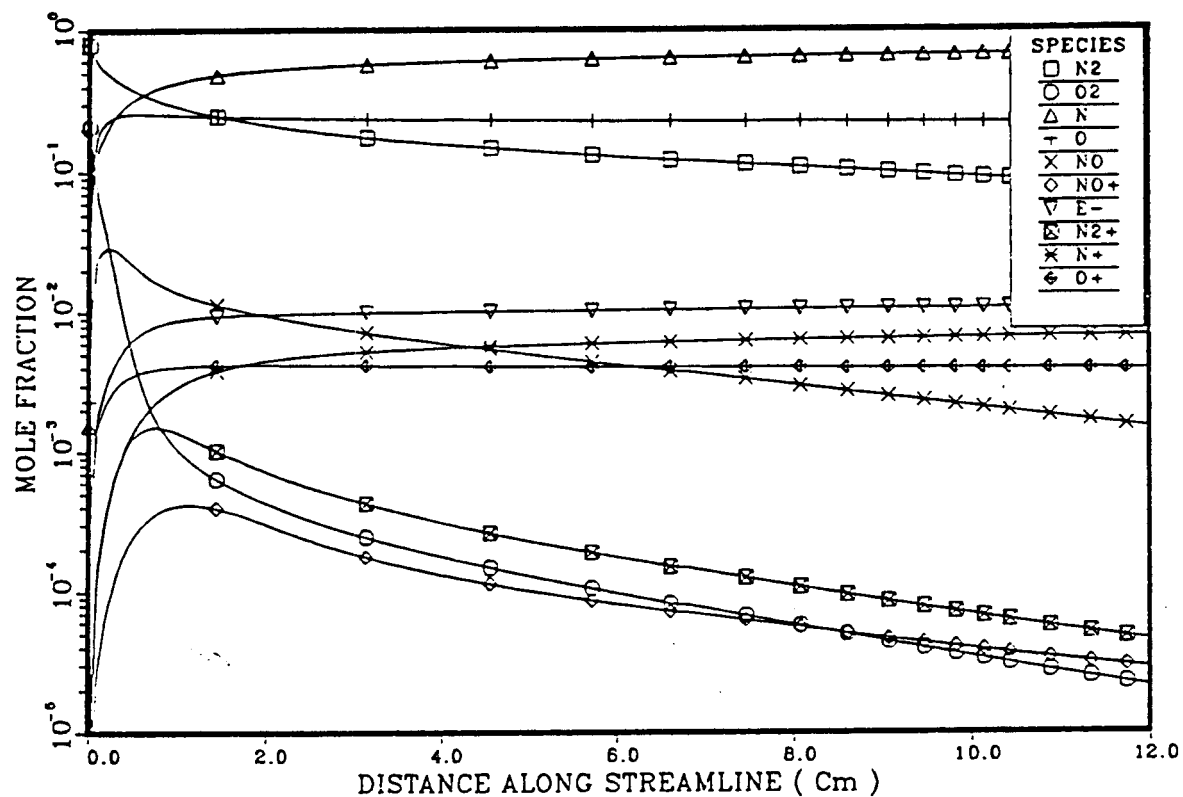


46(c)

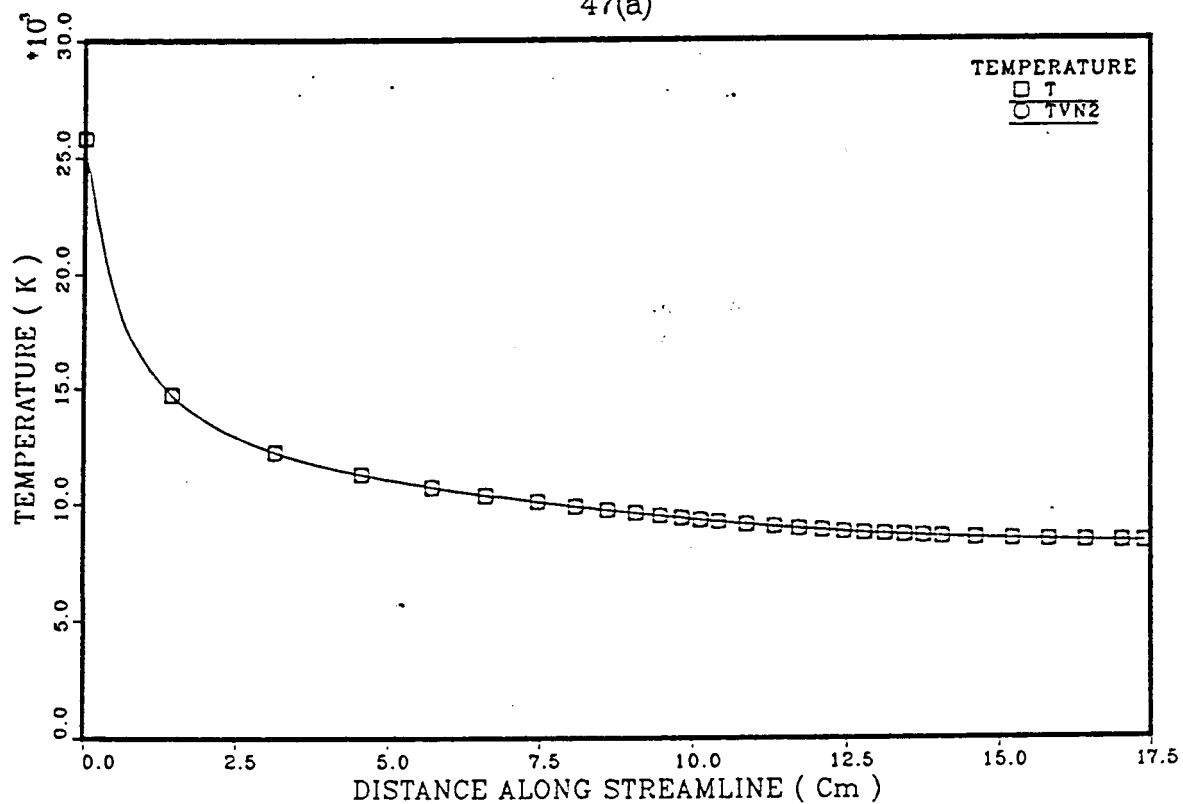


46(d)

FIGURES 46(a),46(b),46(c),46(d).COORD,V=10 Km/s, RR2

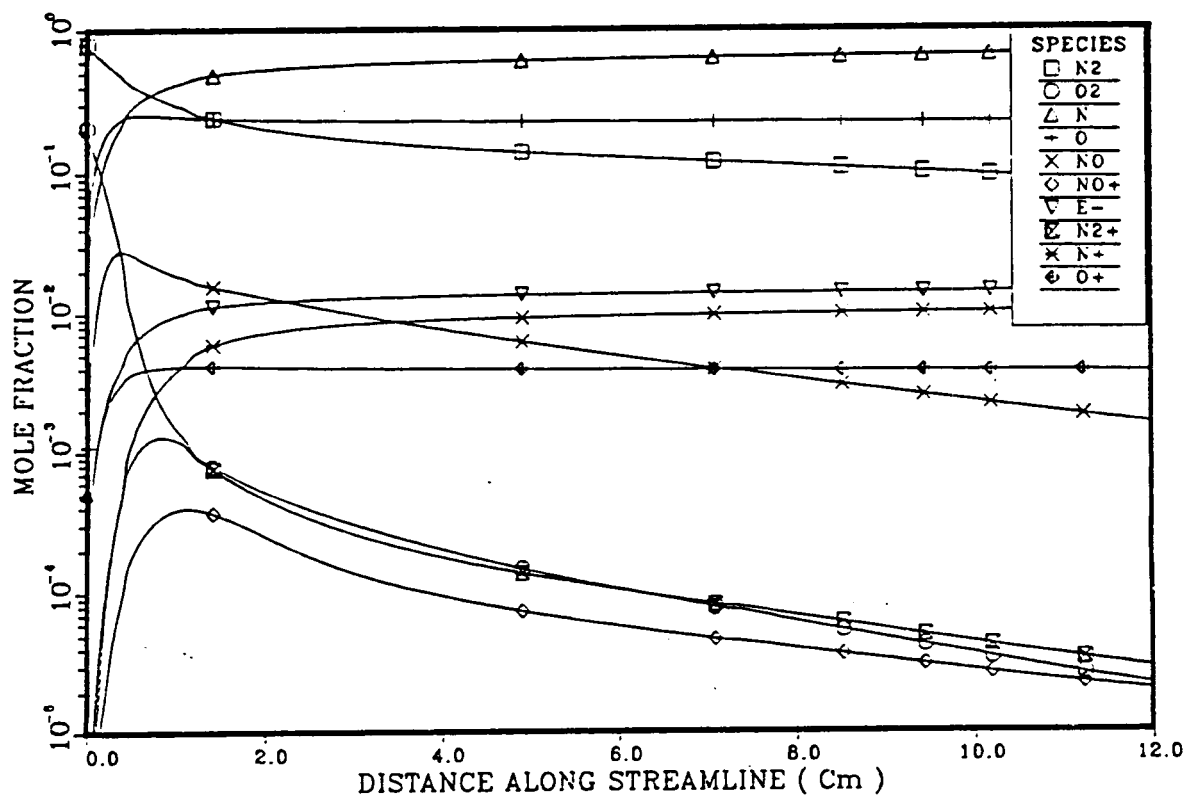


47(a)

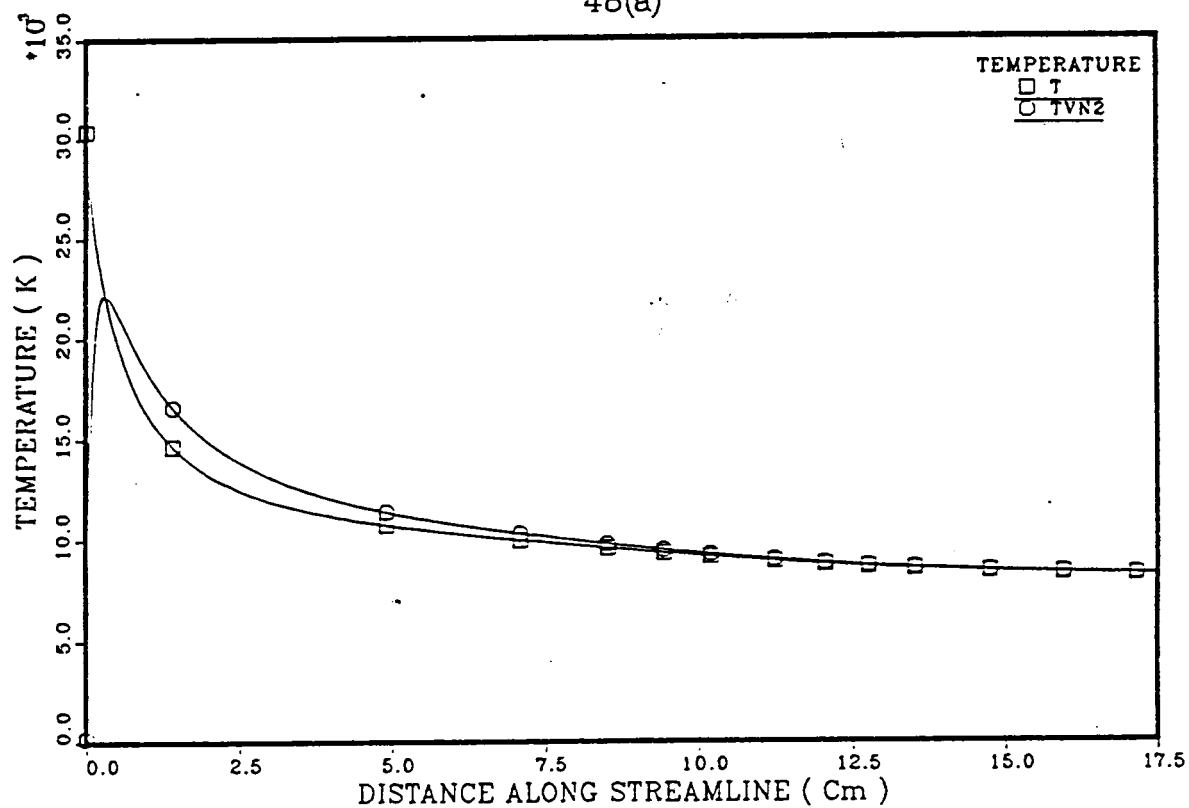


47(b)

FIGURES 47(a),47(b).VEQ MODEL AT V=8.9 Km/s, RR2

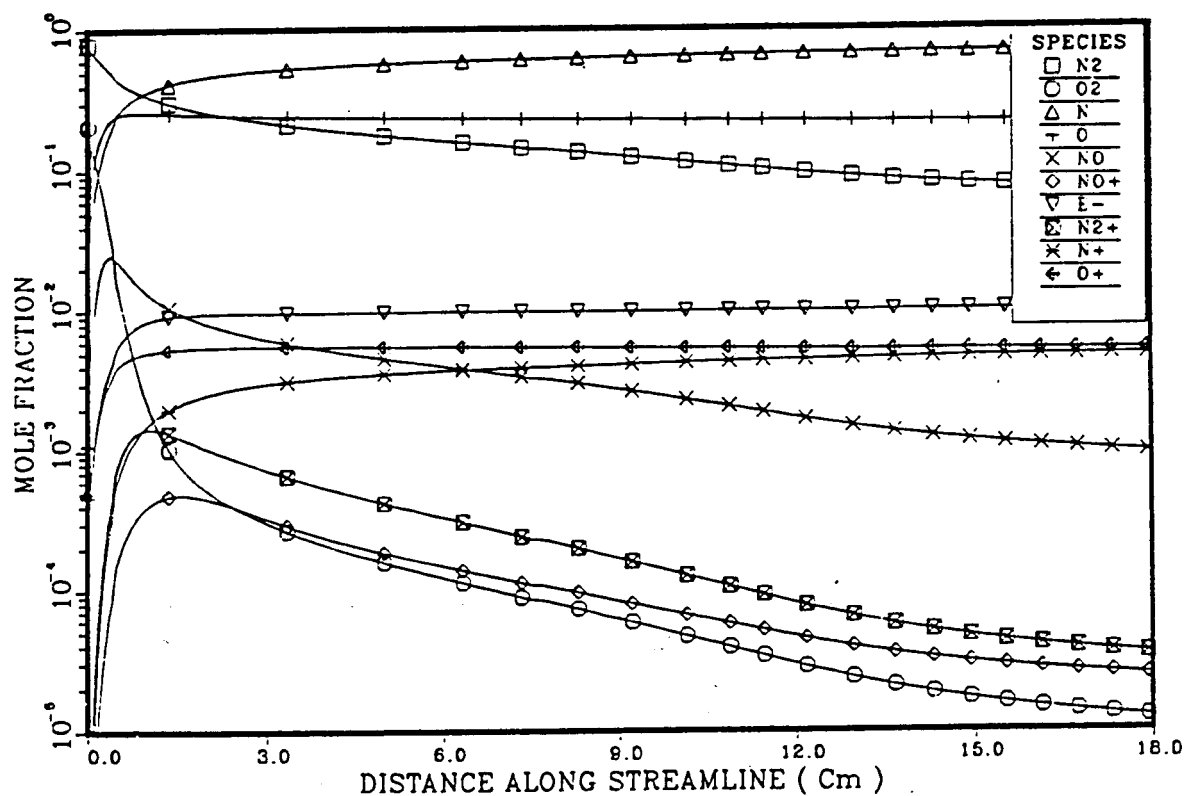


48(a)

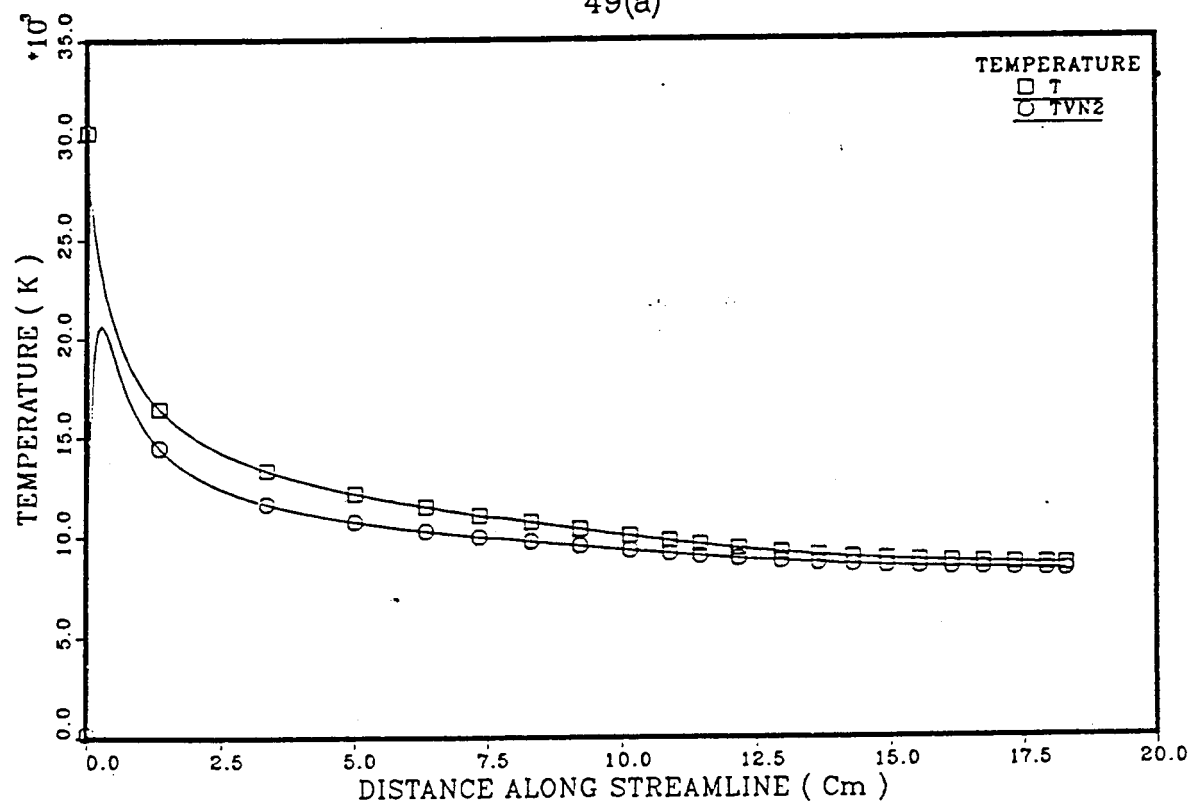


48(b)

FIGURES 48(a),48(b).CVD MODEL AT V=8.9 Km/s, RR2

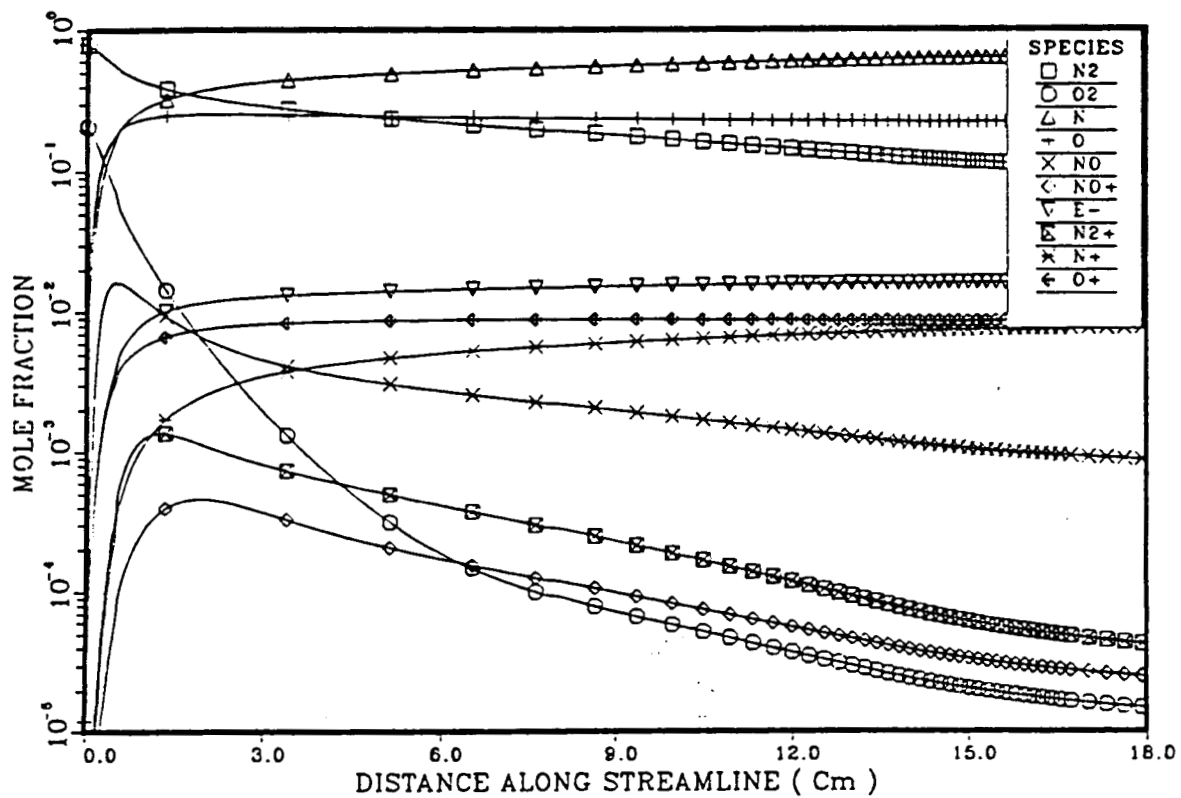


49(a)

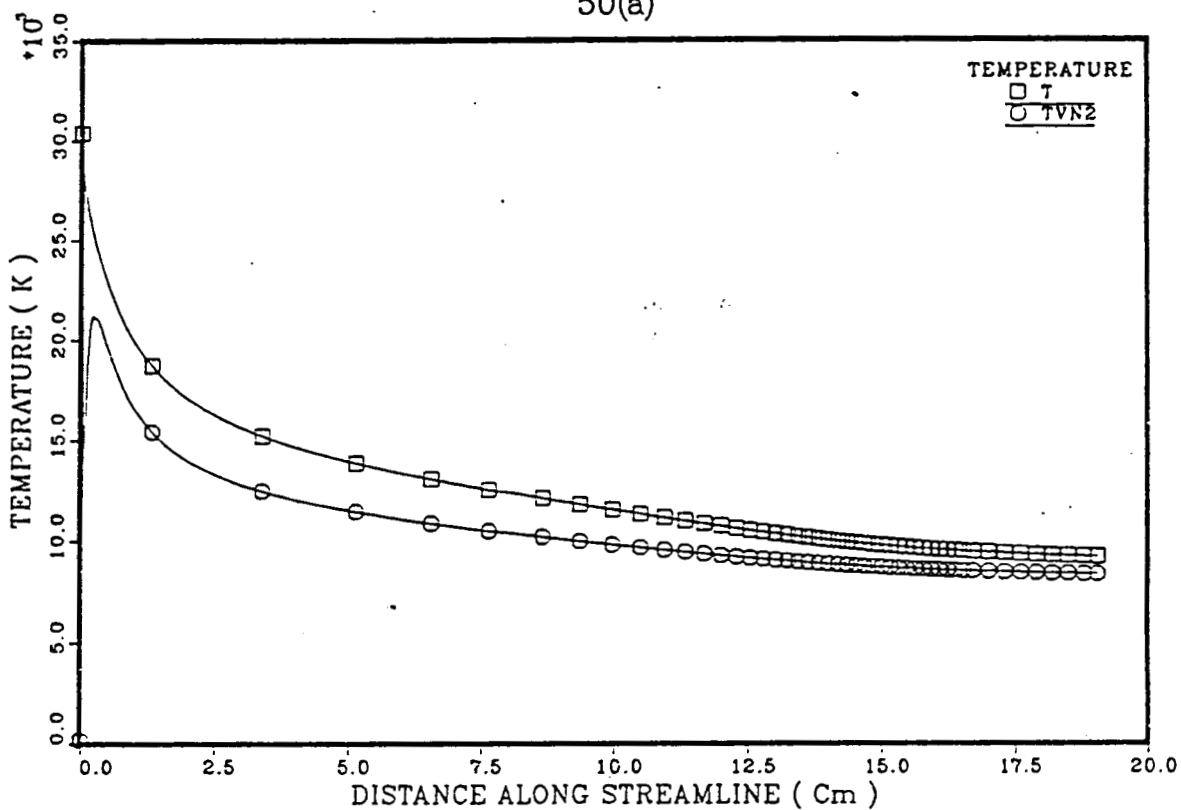


49(b)

FIGURES 49(a),49(b).CVDV MODEL AT V=8.9 Km/s, RR2

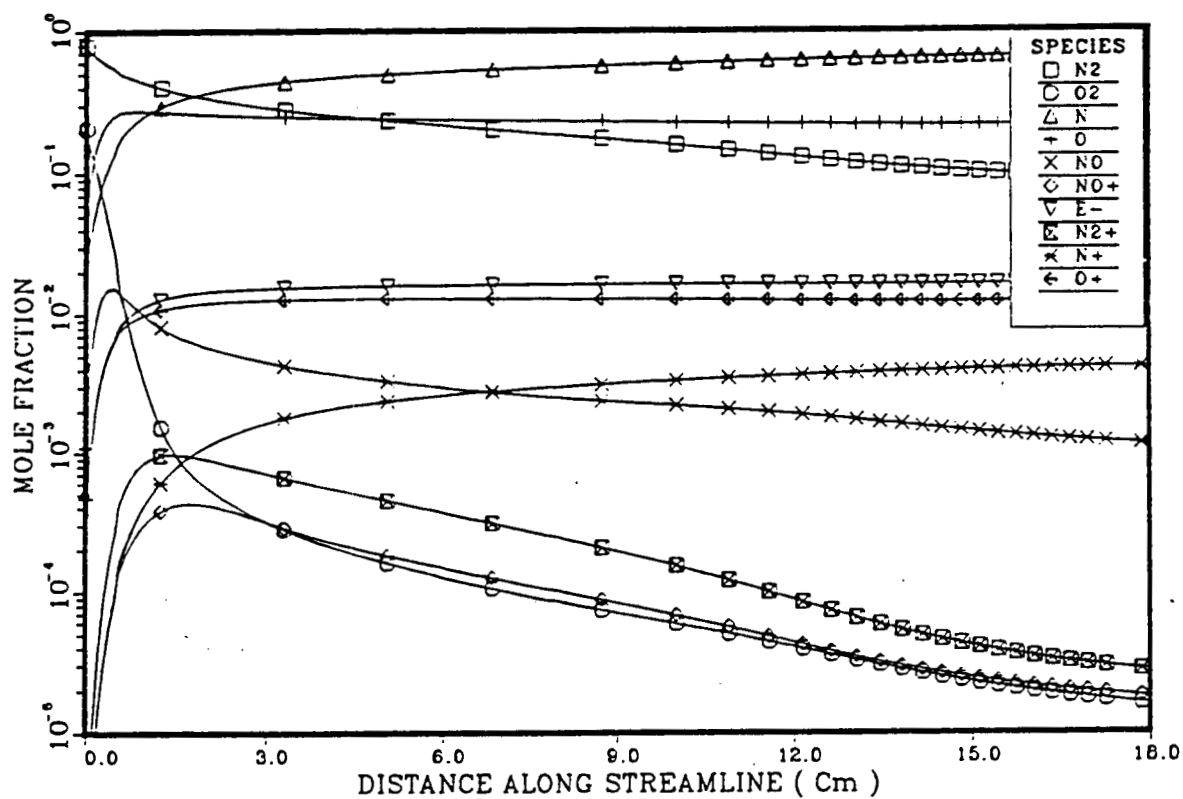


50(a)

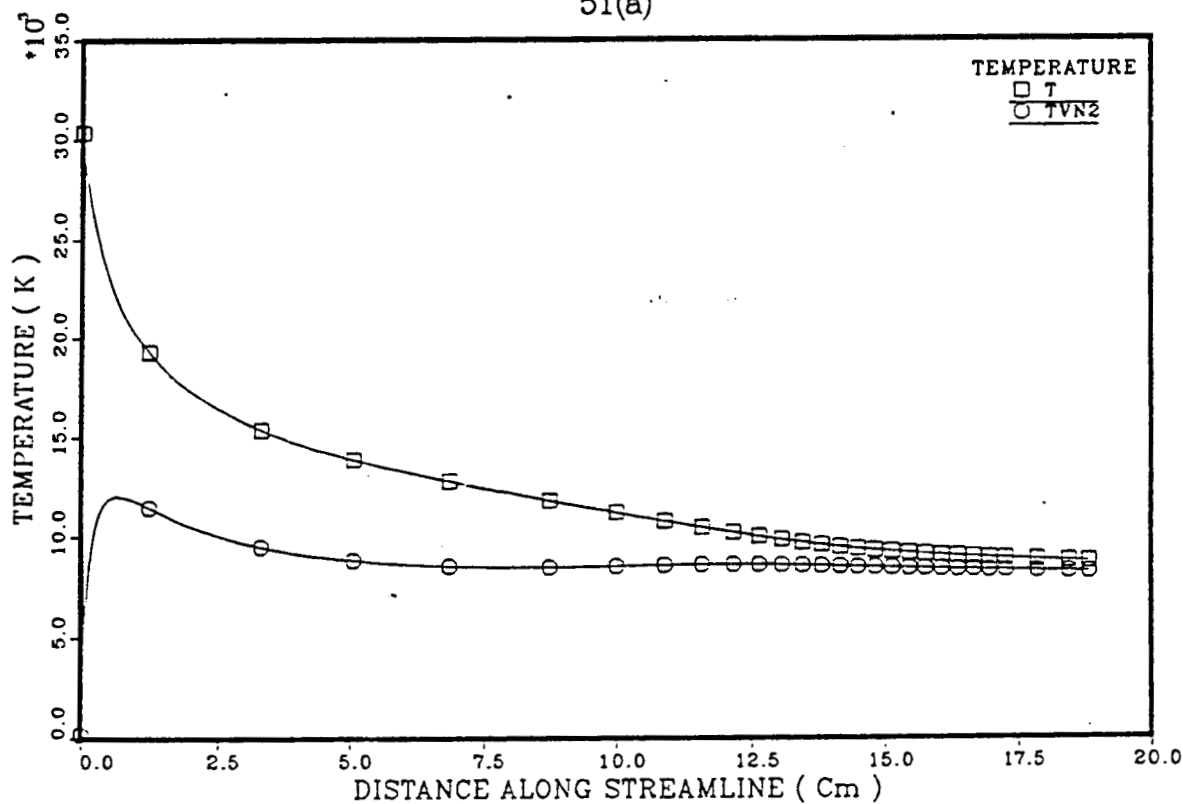


50(b)

FIGURES 50(a),50(b).CVDV-P MODEL AT V=8.9 Km/s, RR2

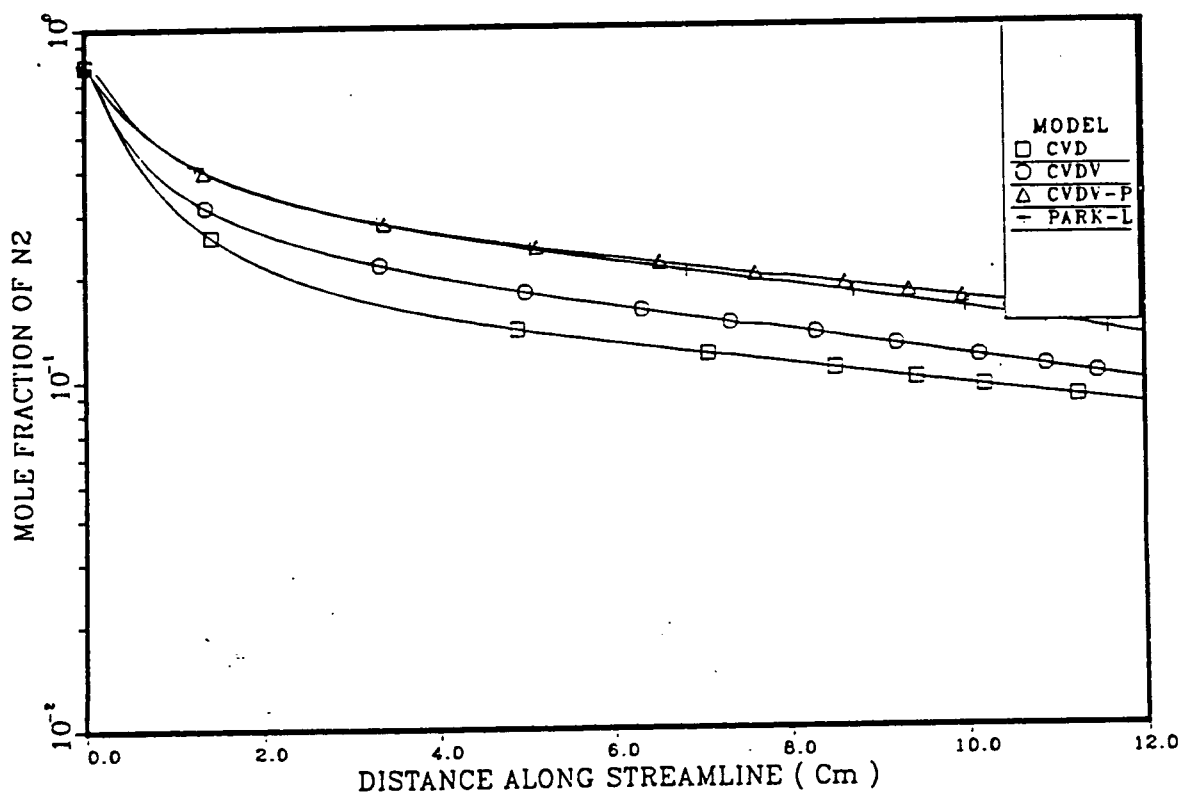


51(a)

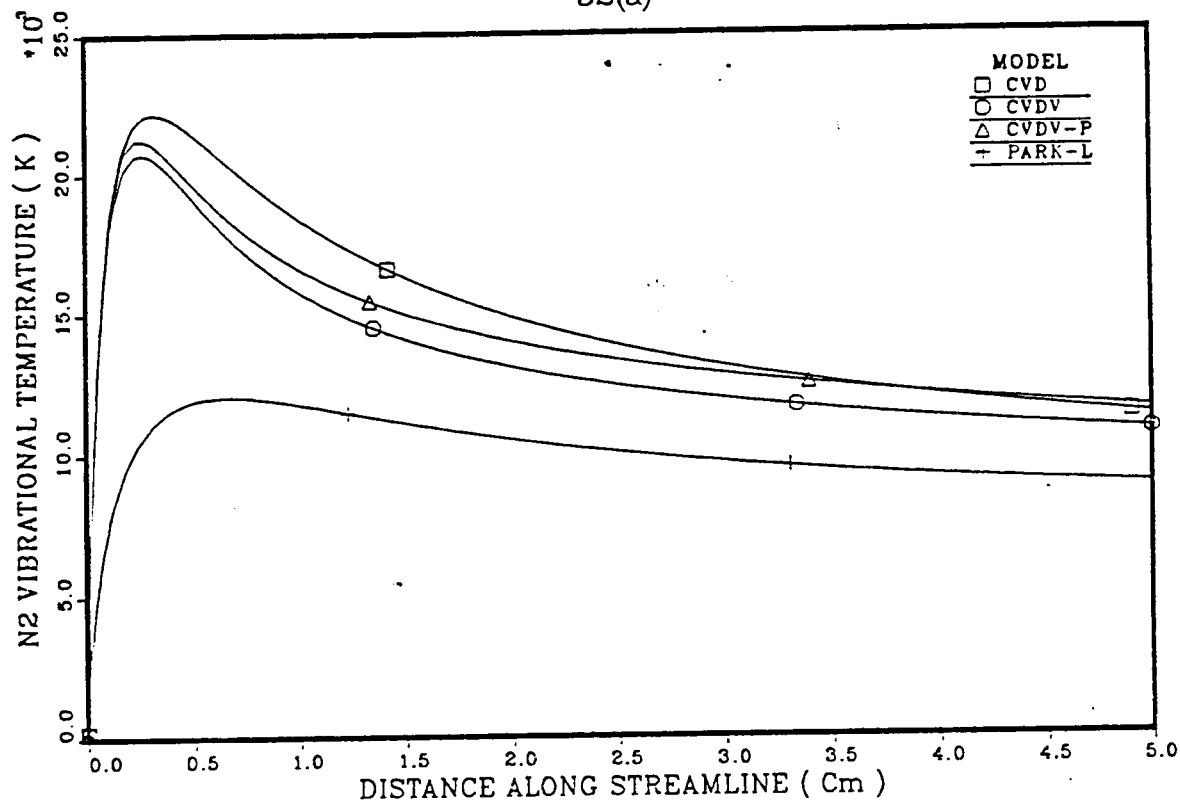


51(b)

FIGURES 51(a),51(b).PARK-L MODEL AT V=8.9 Km/s, RR2

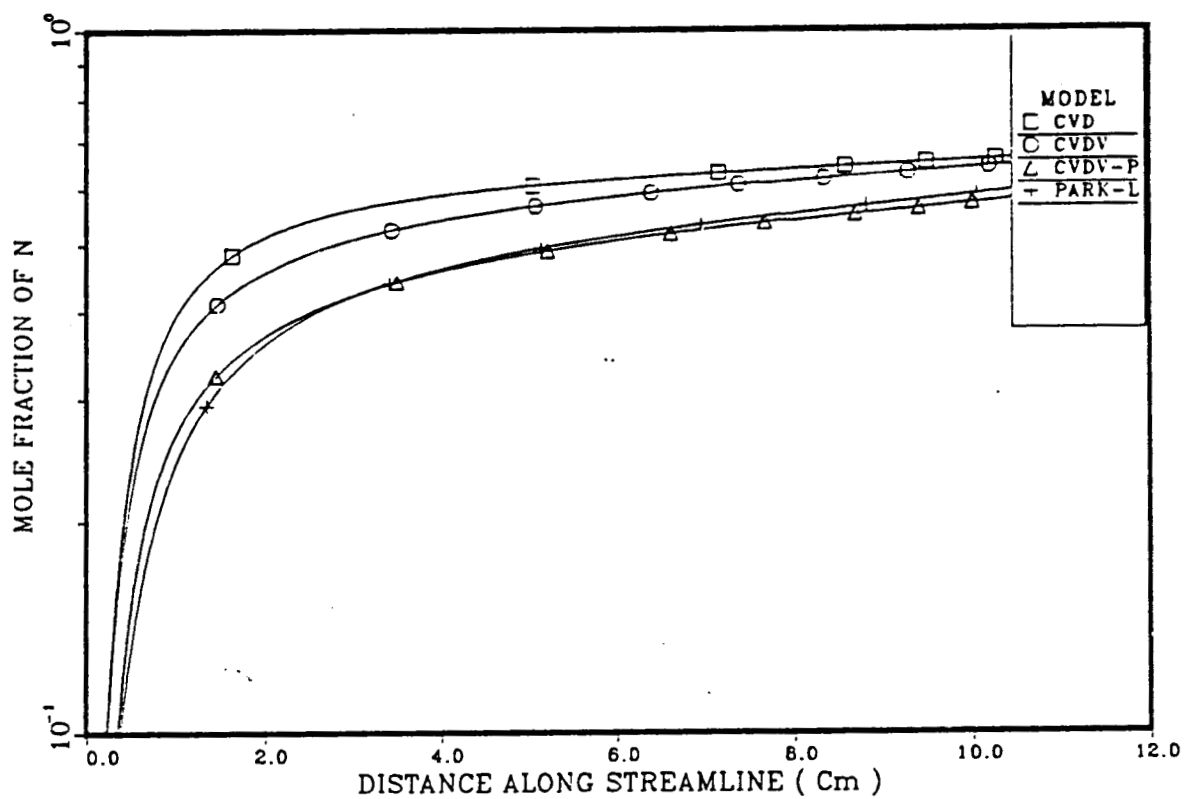


52(a)

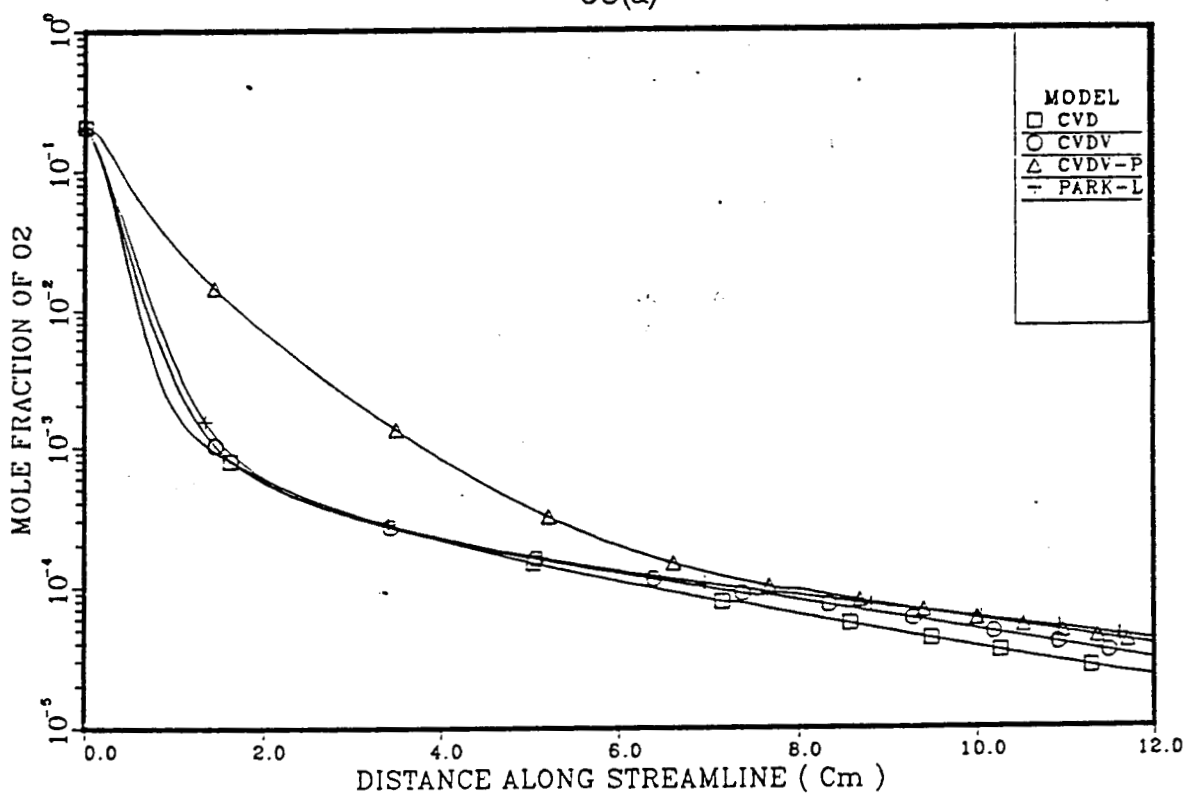


52(b)

FIGURES 52(a),52(b).PROFILES AT V=8.9 Km/s, RR2

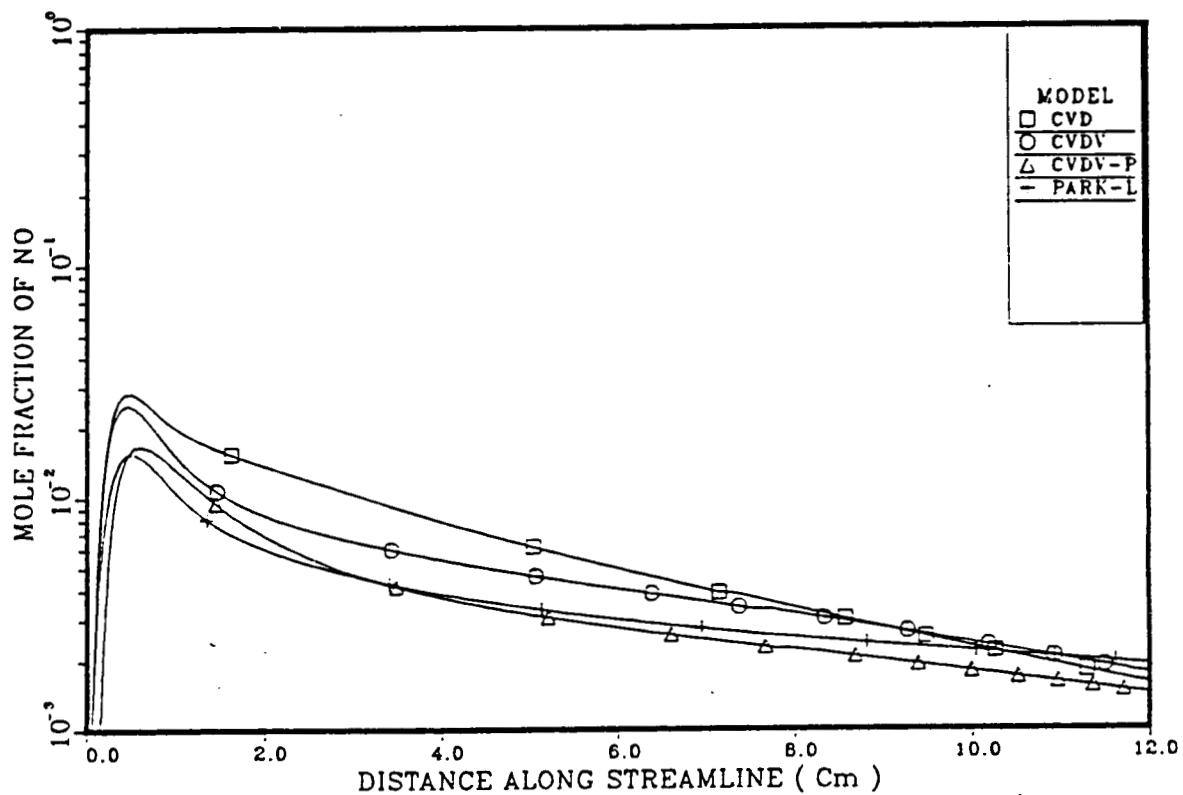


53(a)

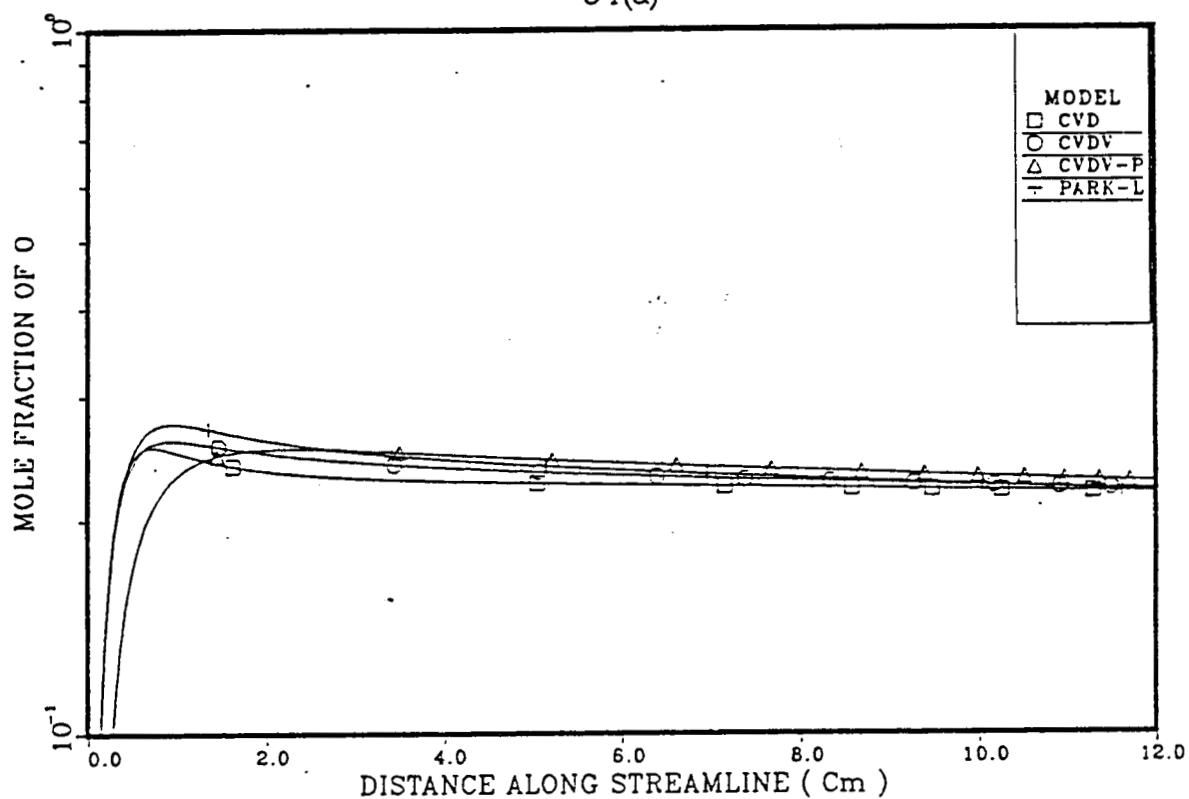


53(b)

FIGURES 53(a),53(b).PROFILES AT V=8.9 Km/s, RR2

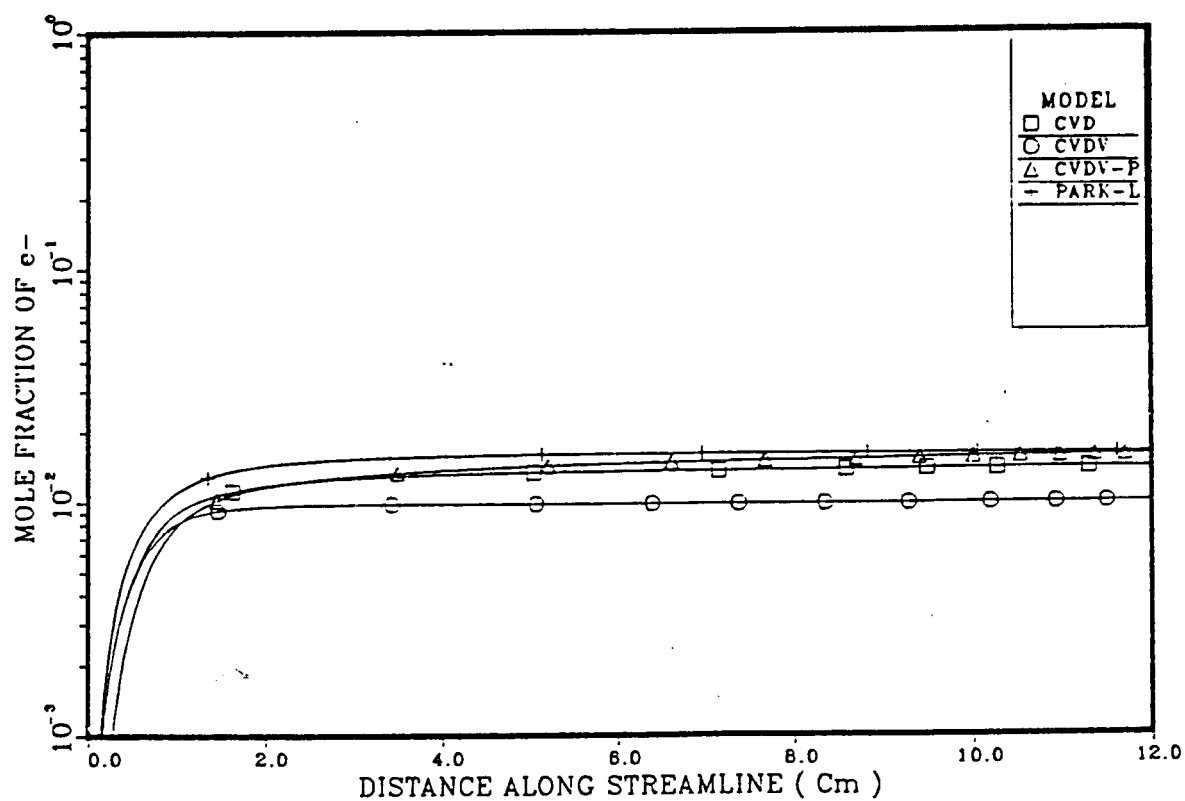


54(a)

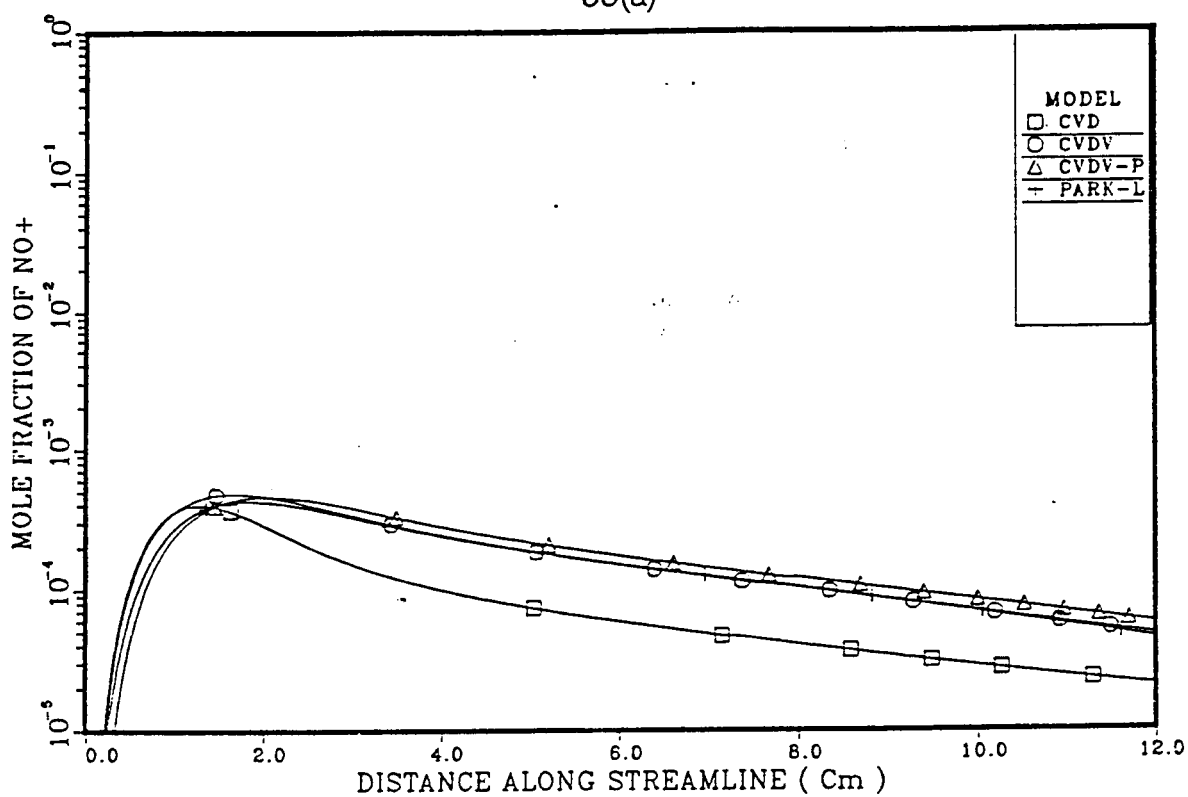


54(b)

FIGURES 54(a),54(b).PROFILES AT V=8.9 Km/s, RR2

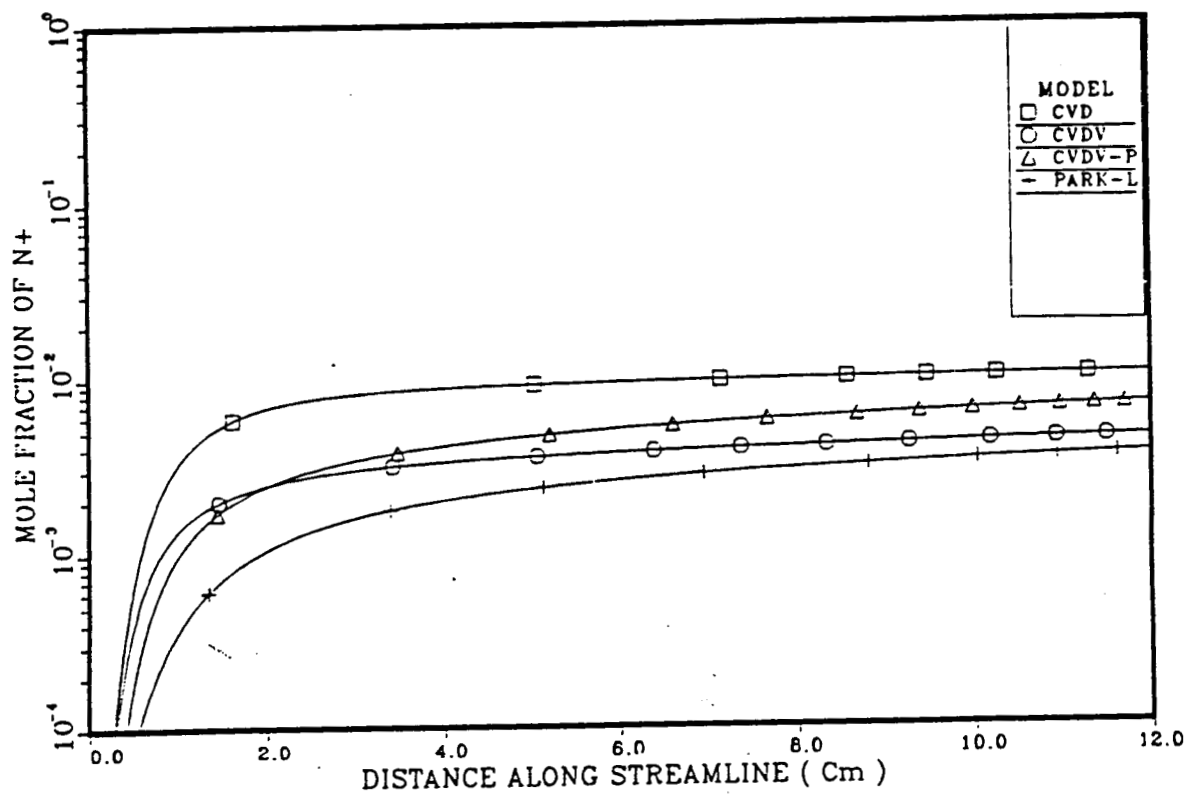


55(a)

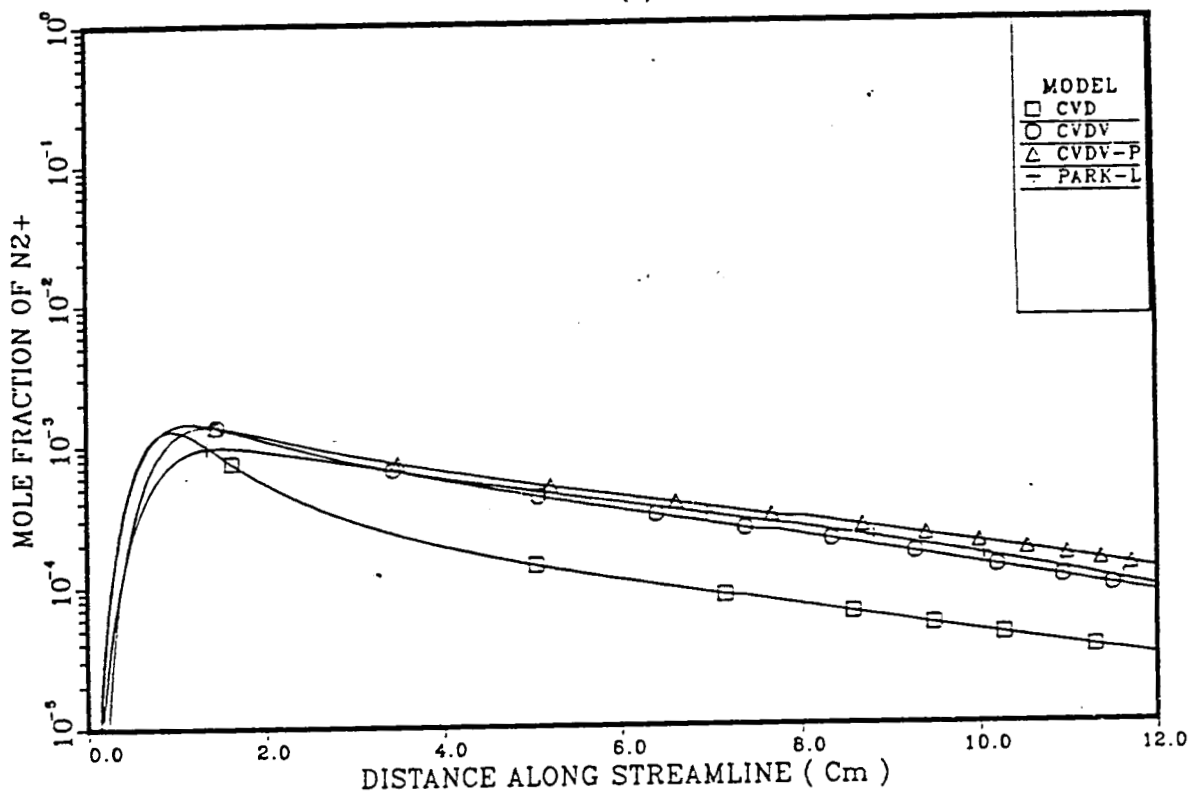


55(b)

FIGURES 55(a),55(b).PROFILES AT V=8.9 Km/s, RR2



56(a)



56(b)

FIGURES 56(a),56(b).PROFILES AT $V=8.9$ Km/s, RR2

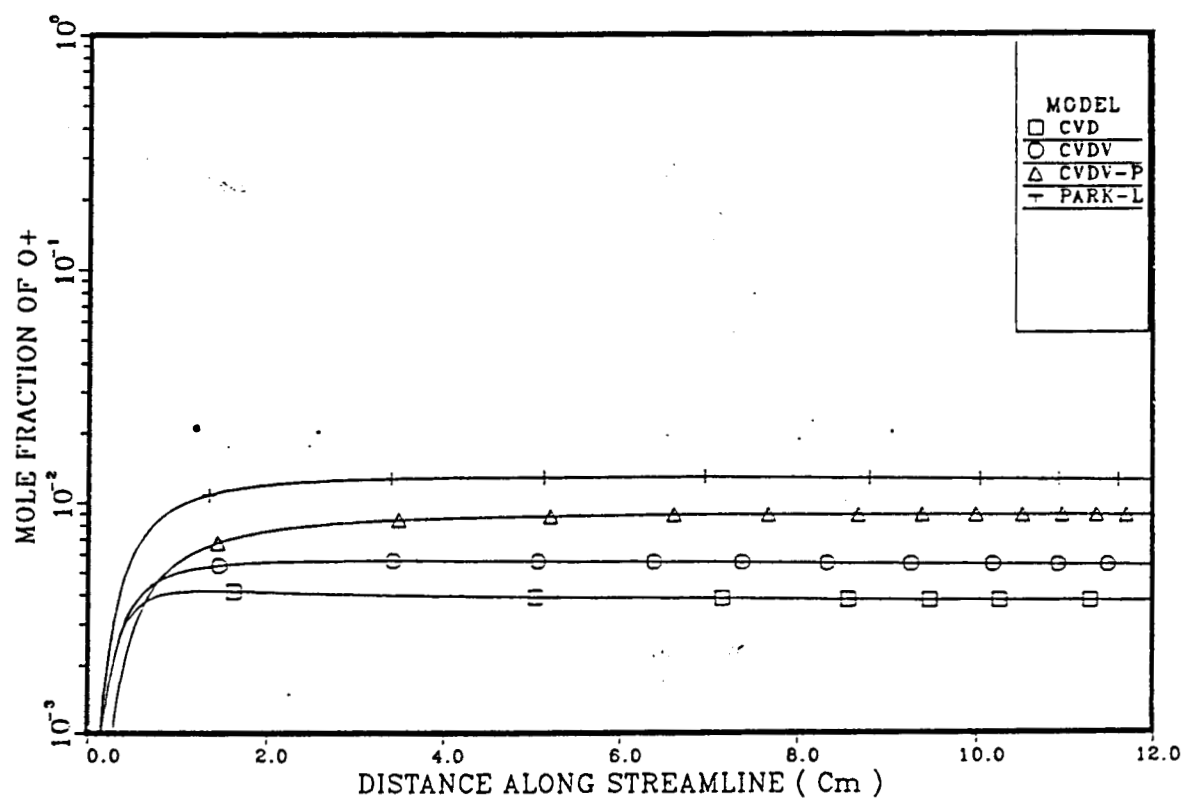
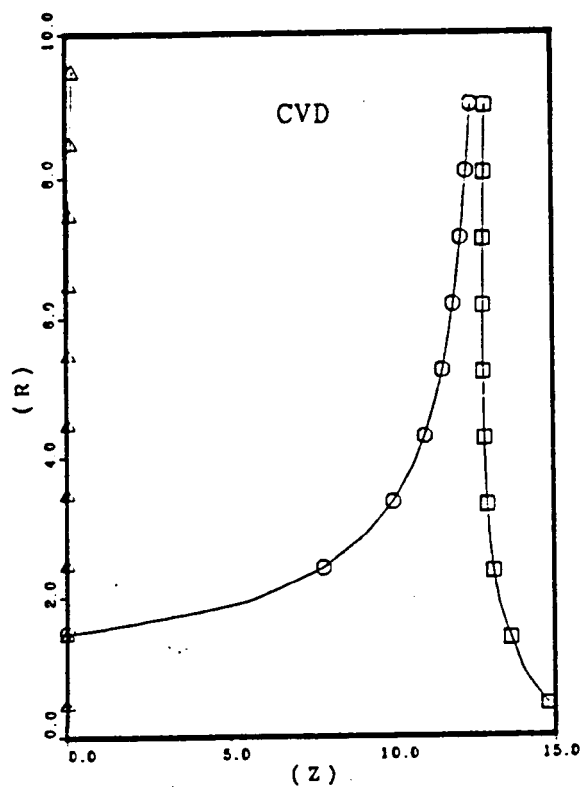
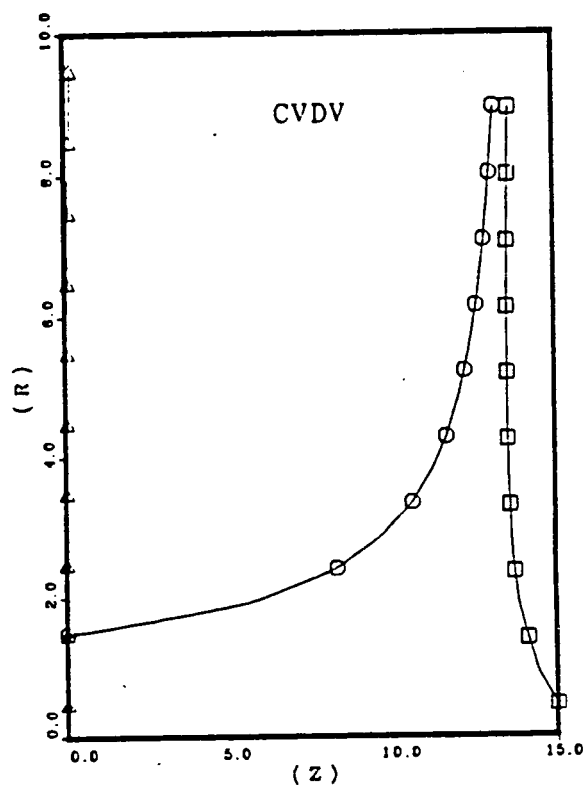


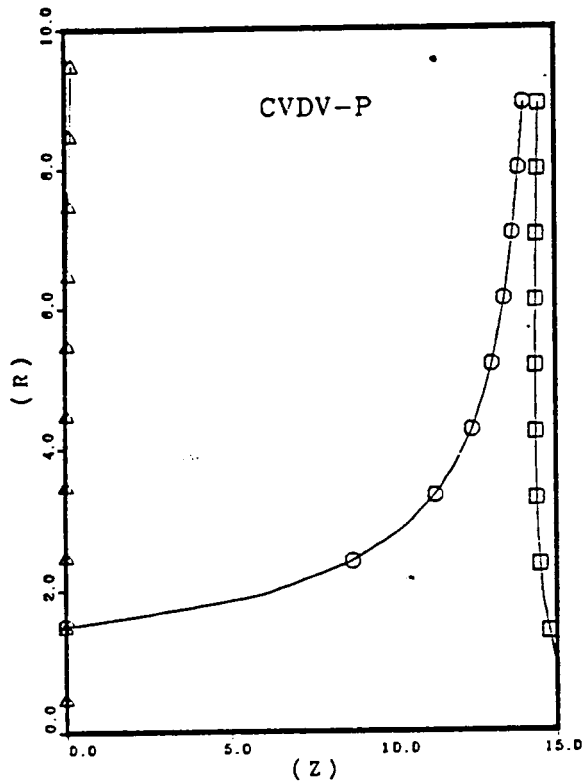
FIGURE 57.PROFILE AT V=8.9 Km/s, RR2



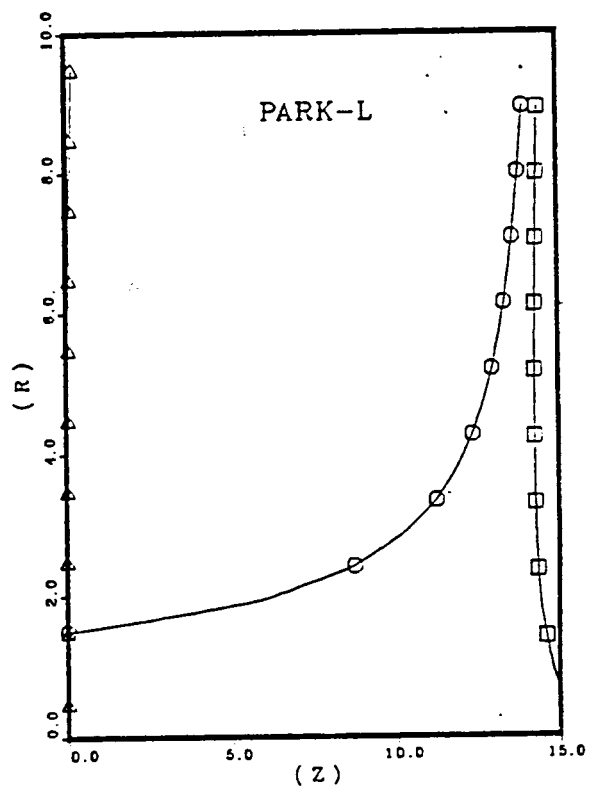
58(a)



58(b)

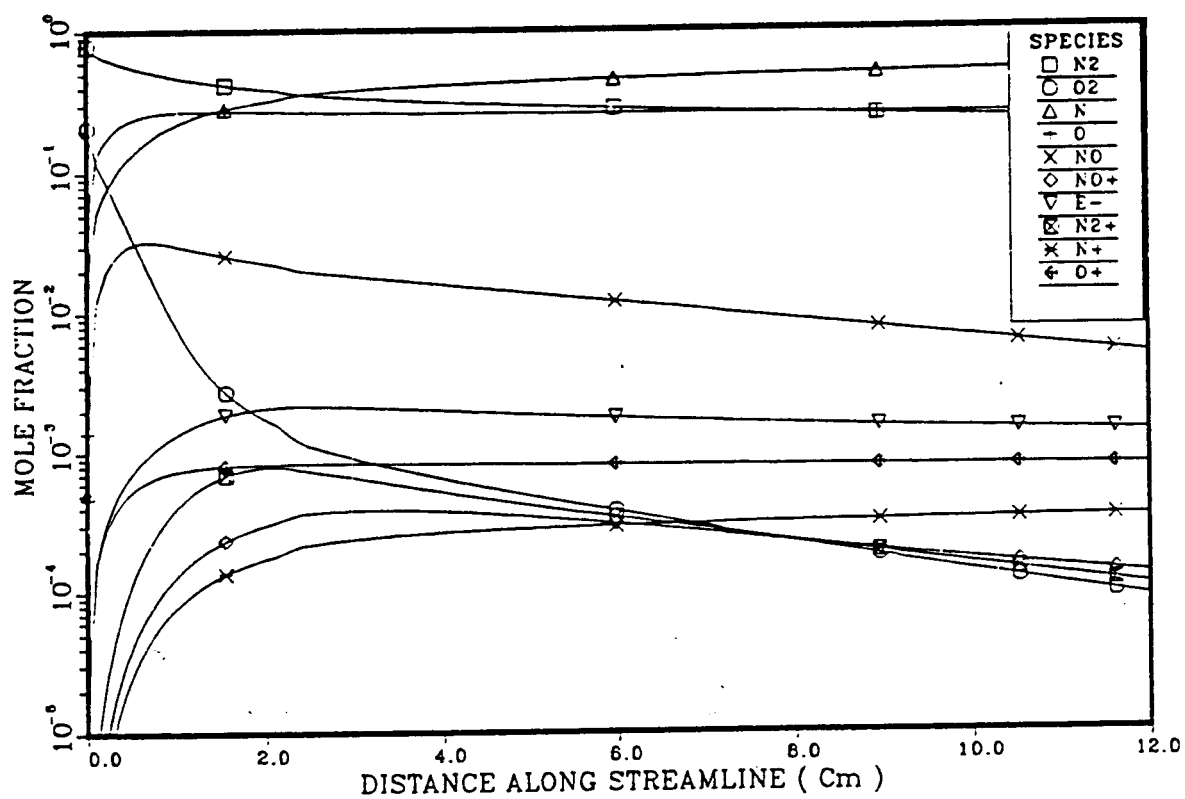


58(c)

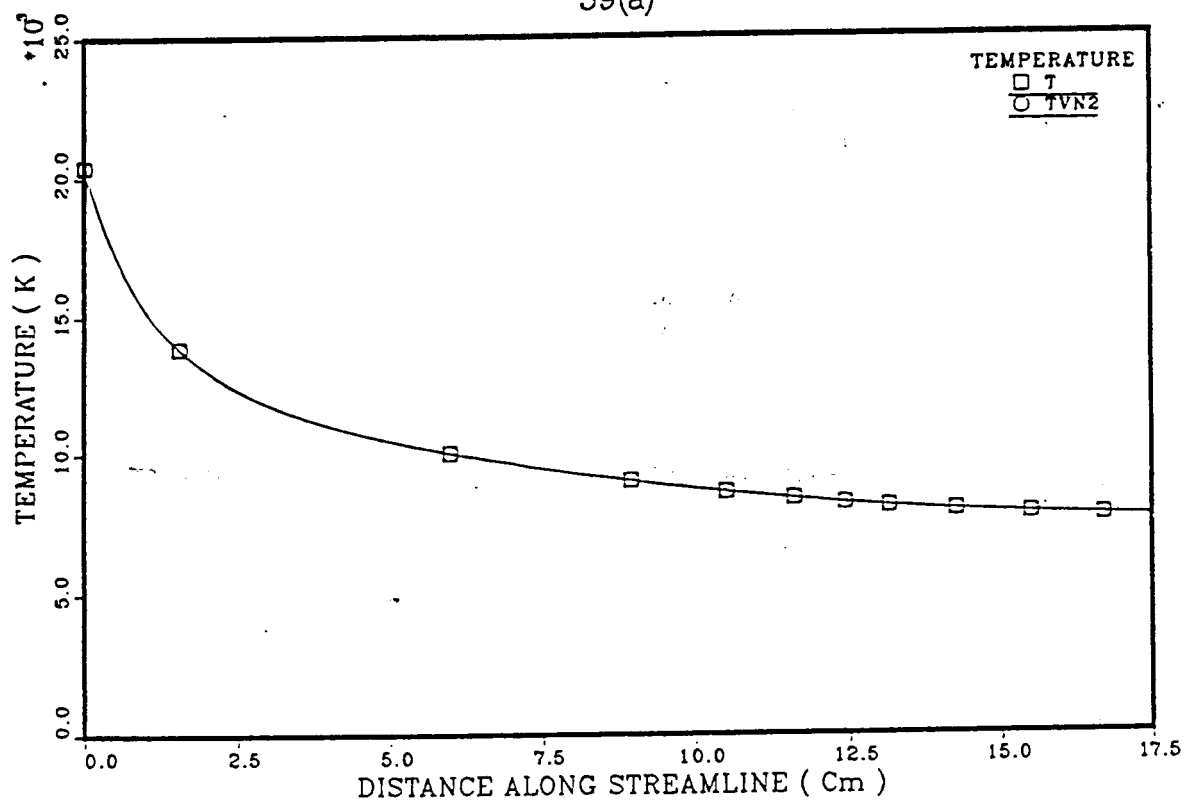


58(d)

FIGURES 58(a),58(b),58(c),58(d).COORD,V=8.9 Km/s, RR2

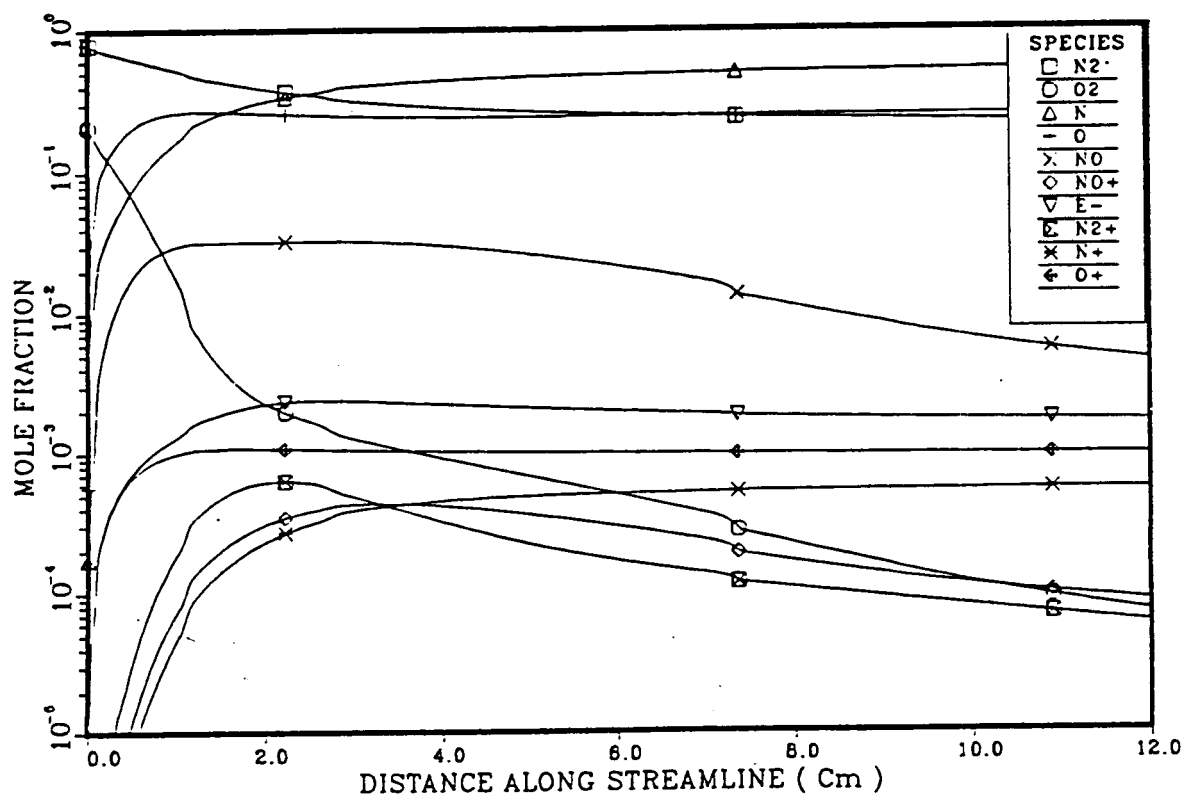


59(a)

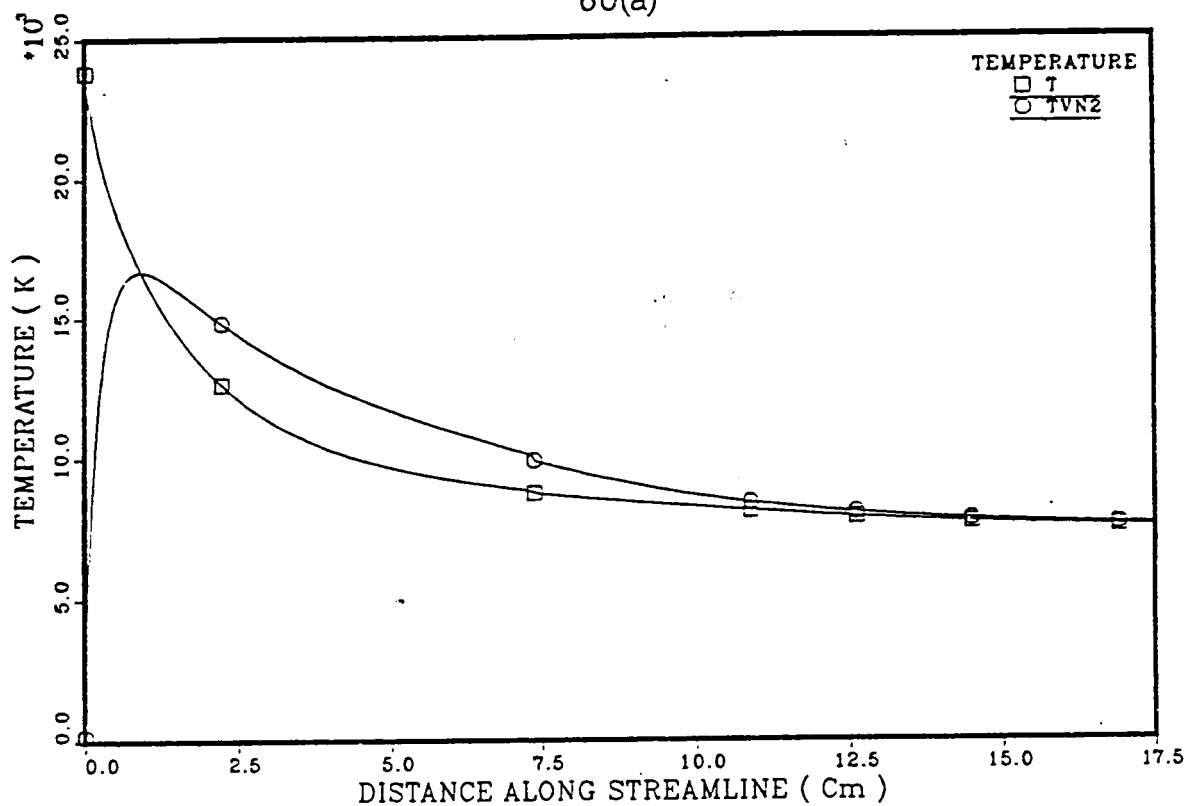


59(b)

FIGURES 59(a),59(b).VEQ MODEL AT $V=7.7$ Km/s, RR2

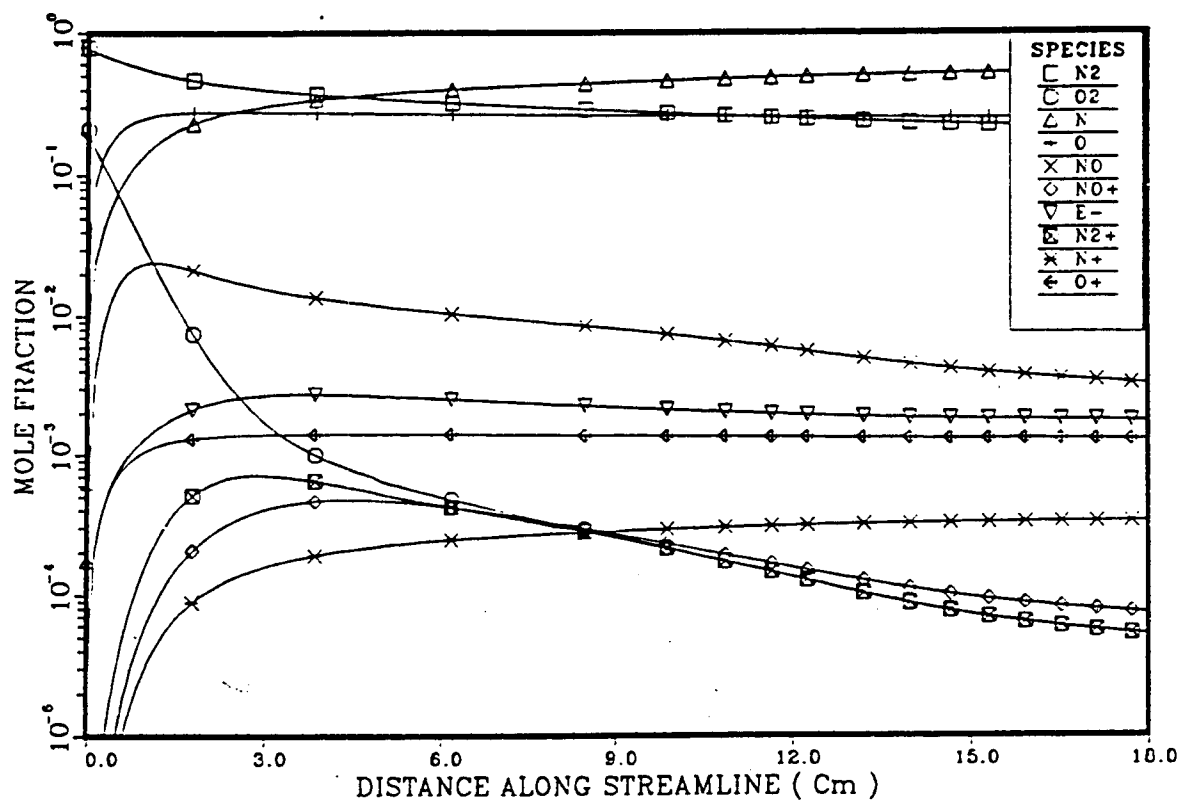


60(a)

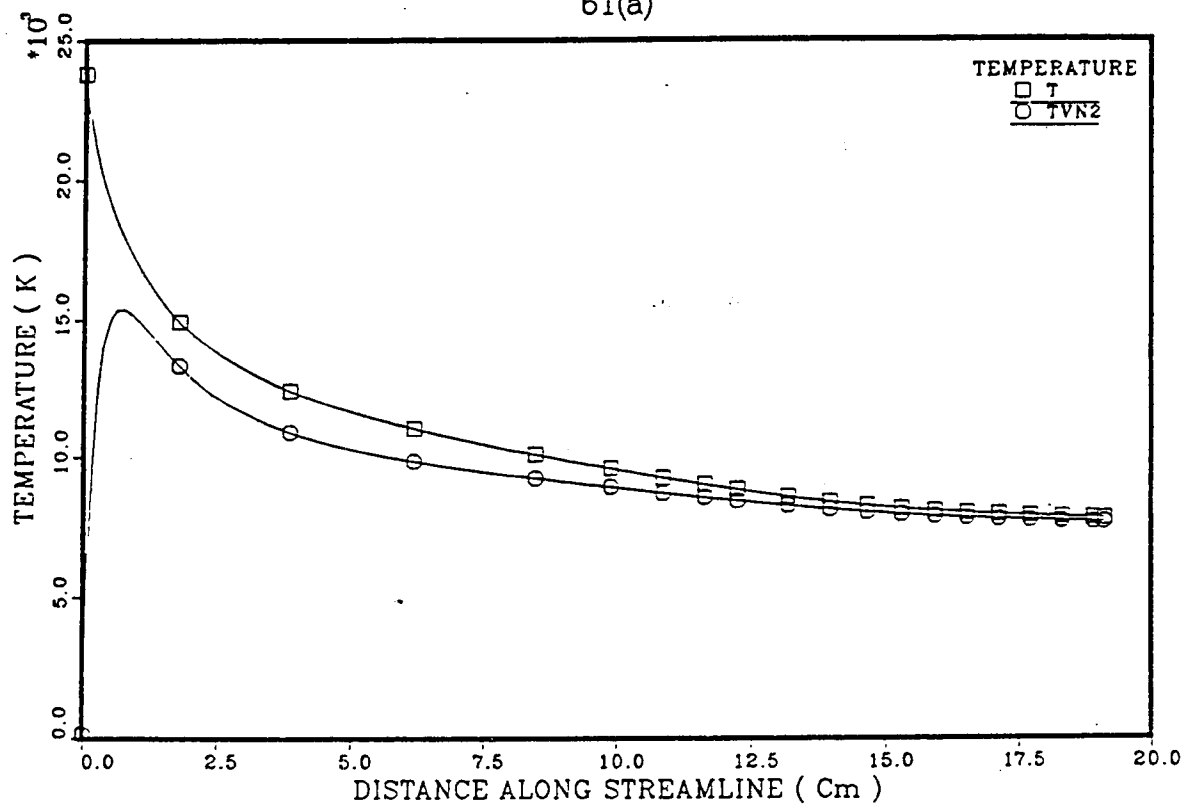


60(b)

FIGURES 60(a),60(b).CVD MODEL AT $V=7.7$ Km/s, RR2

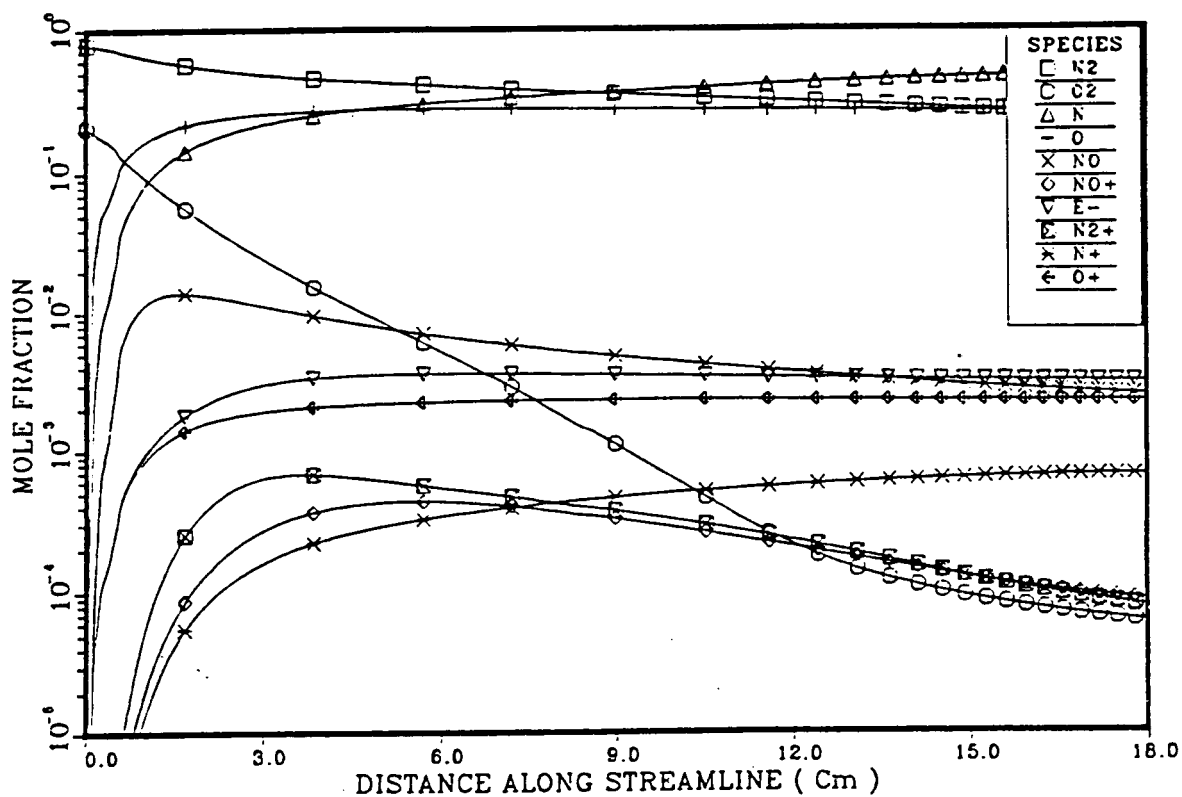


61(a)

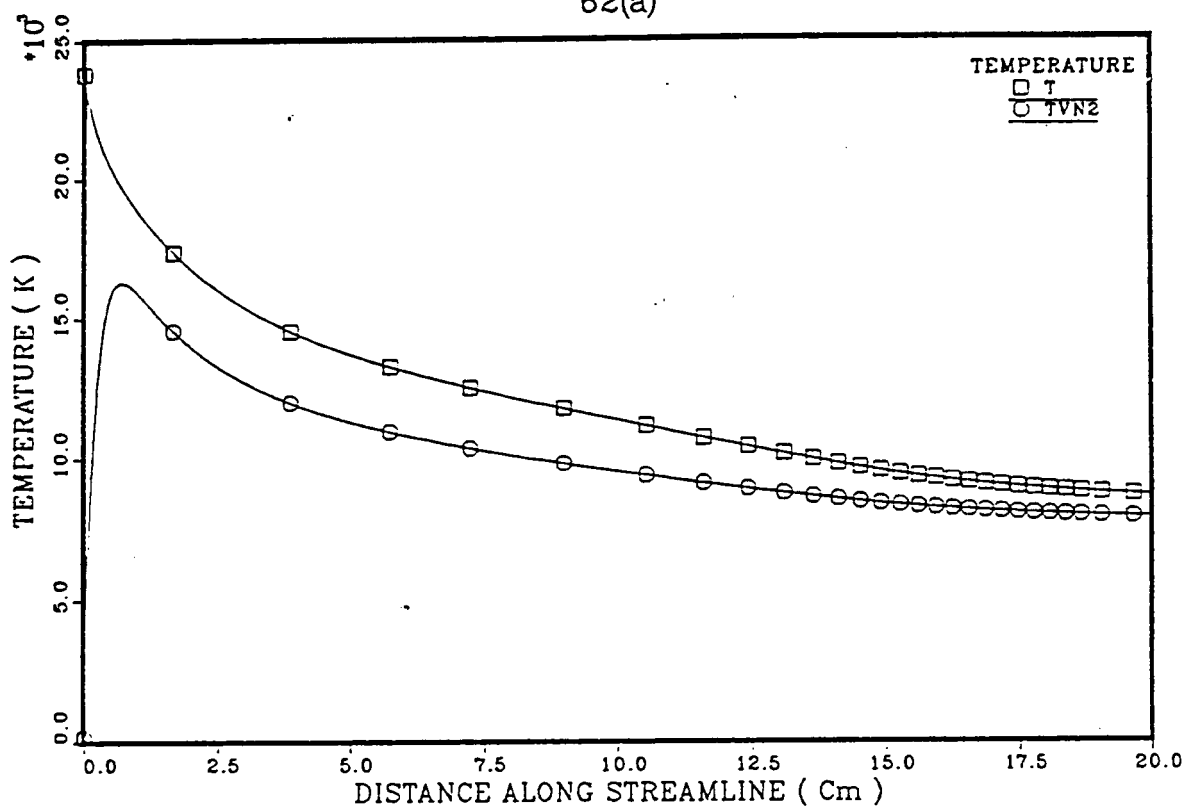


61(b)

FIGURES 61(a),61(b).CVDV MODEL AT V=7.7 Km/s, RR2

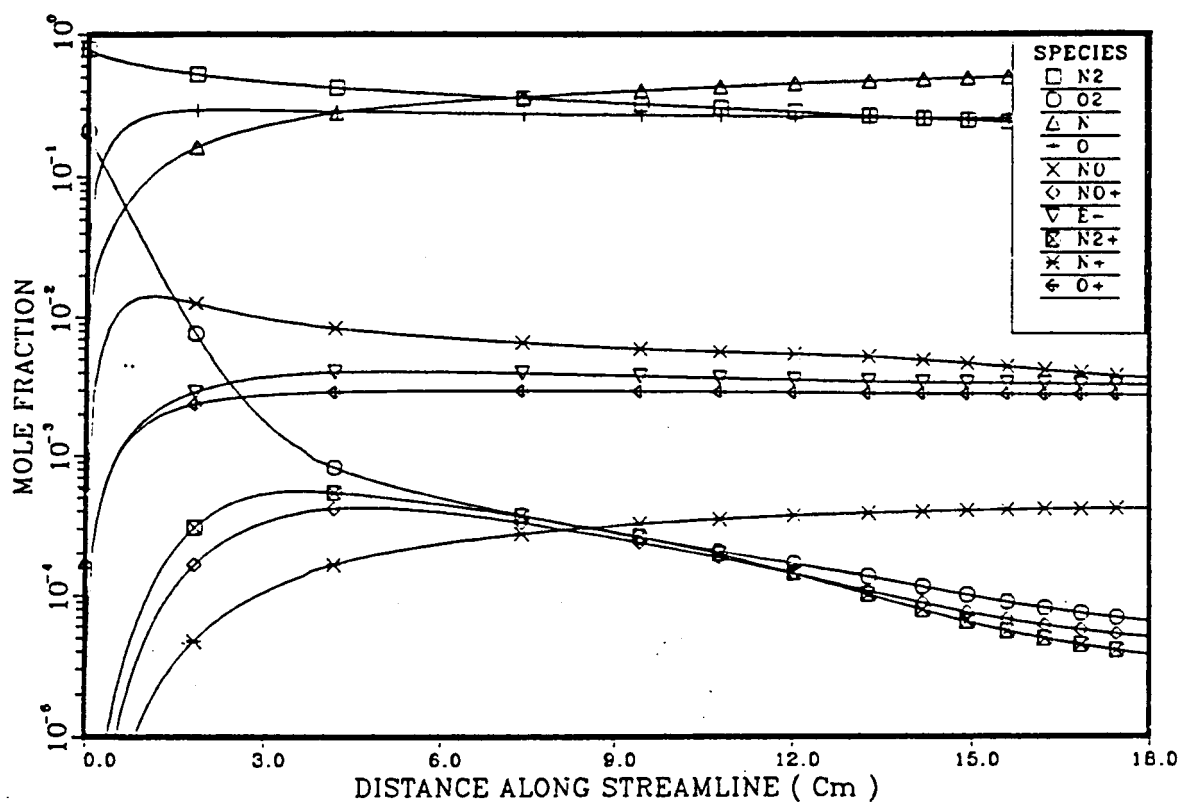


62(a)

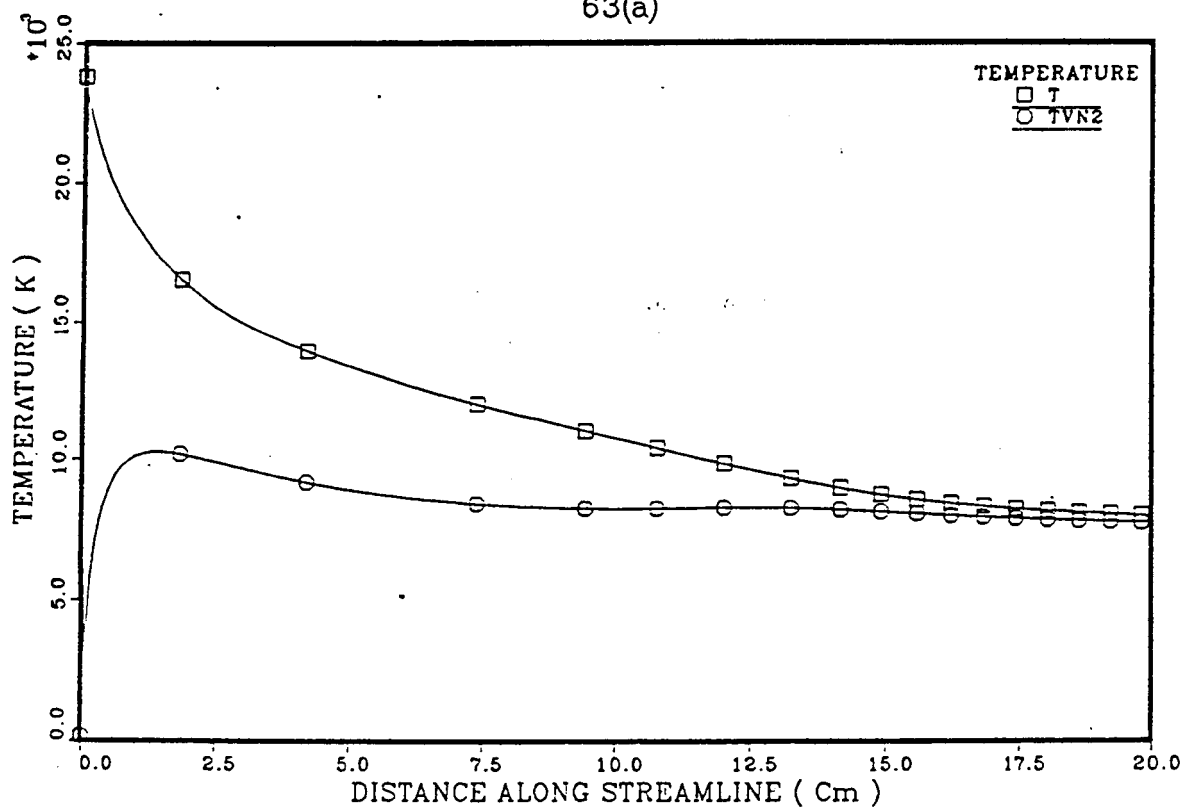


62(b)

FIGURES 62(a),62(b).CVDV-P MODEL AT V=7.7 Km/s, RR2

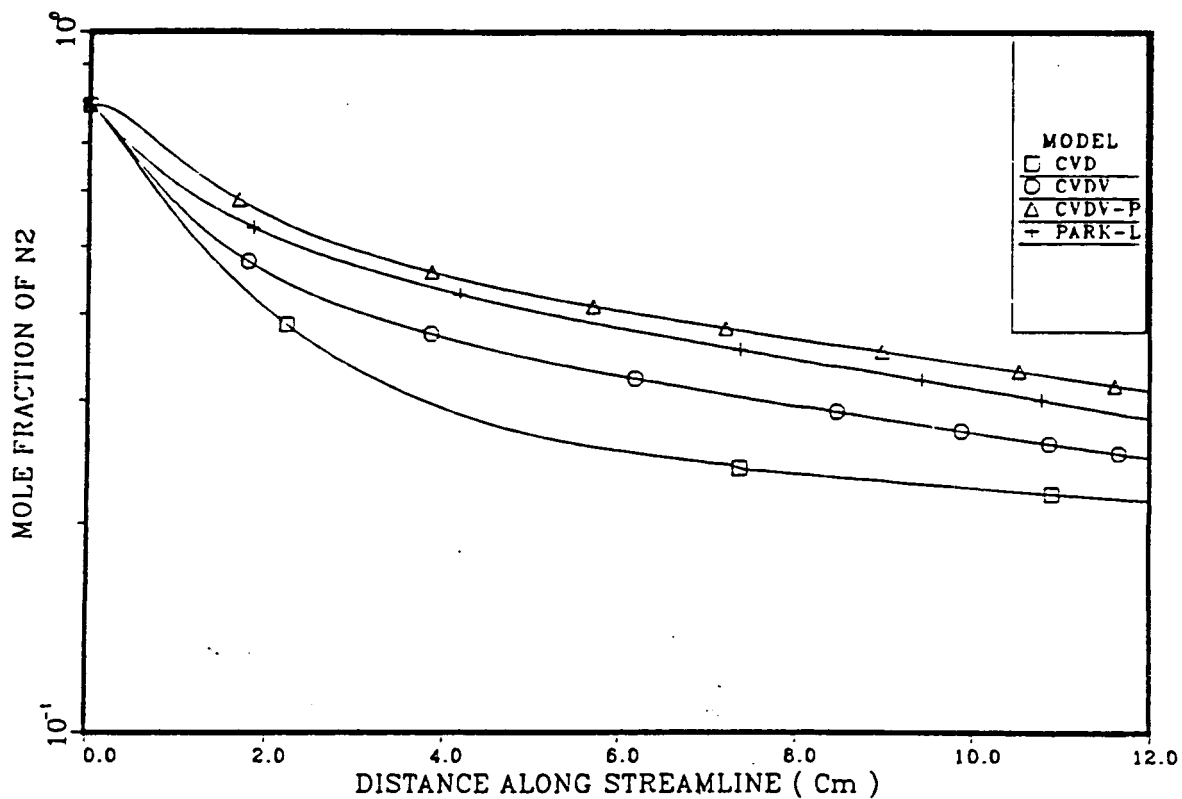


63(a)

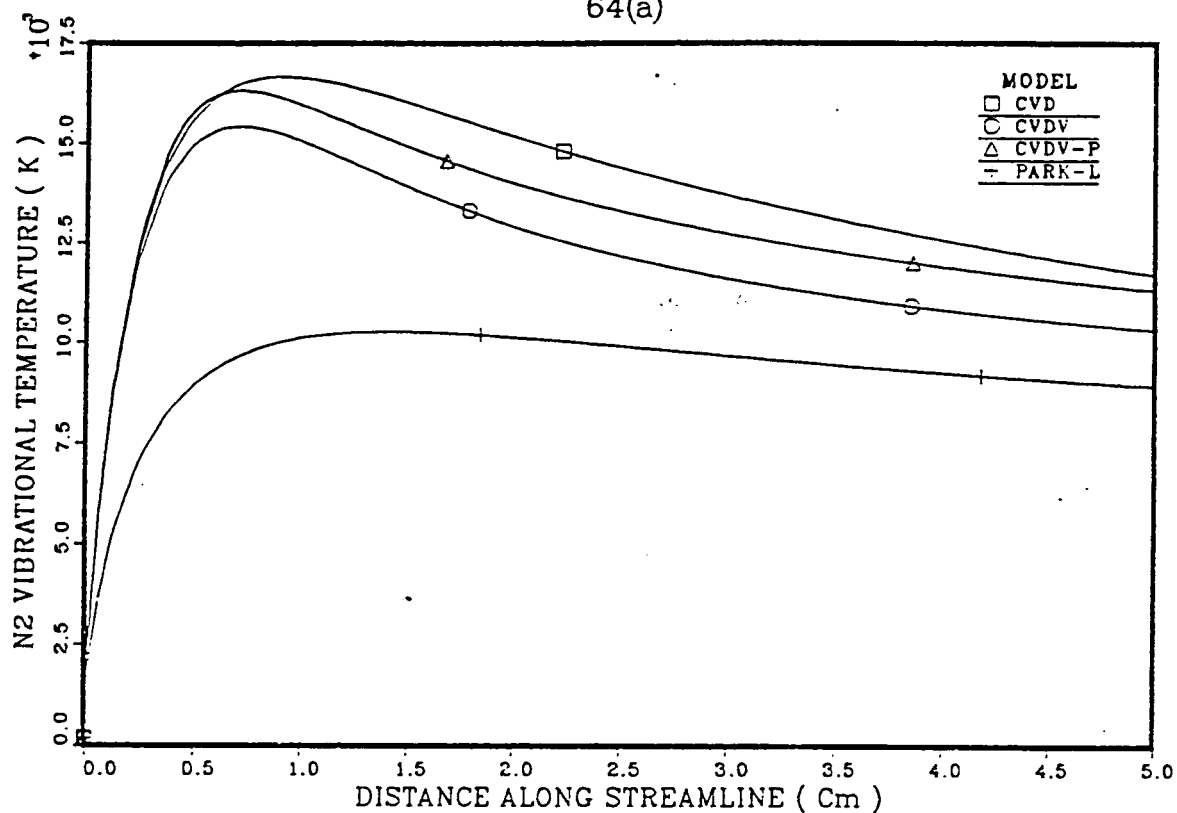


63(b)

FIGURES 63(a),63(b).PARK-L MODEL AT V=7.7 Km/s, RR2

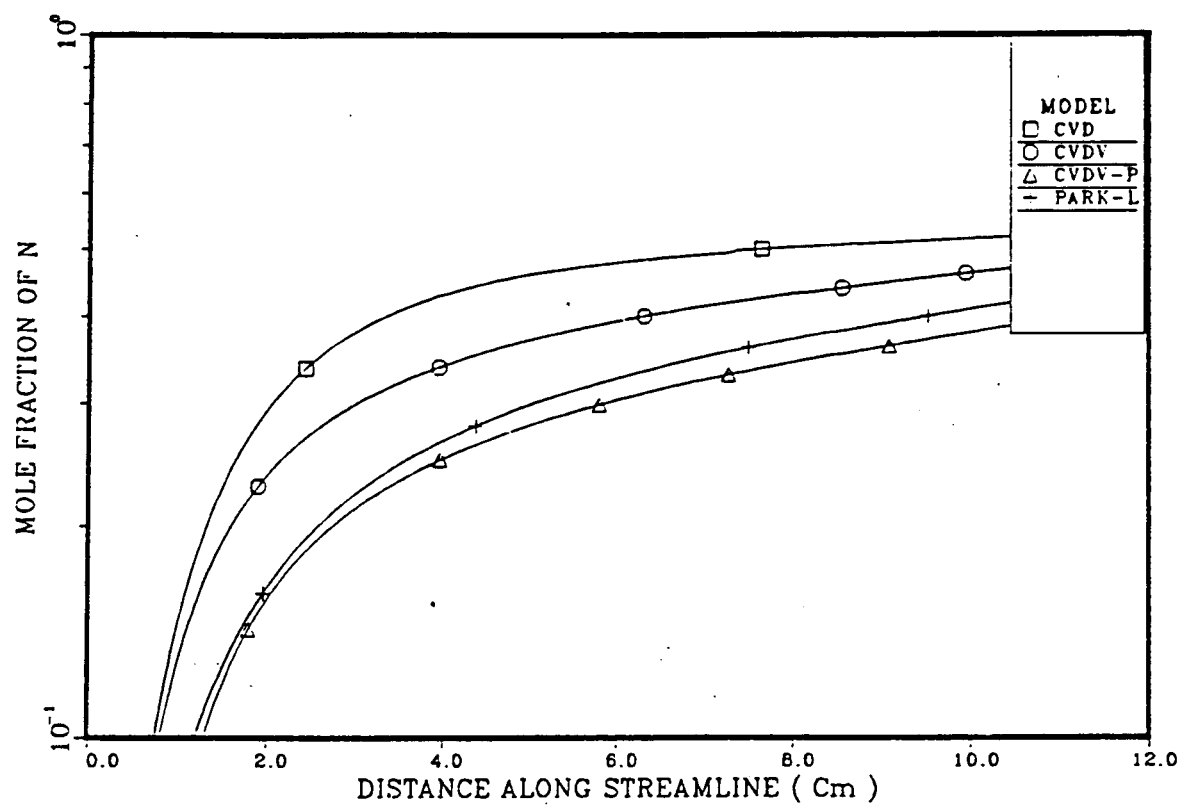


64(a)

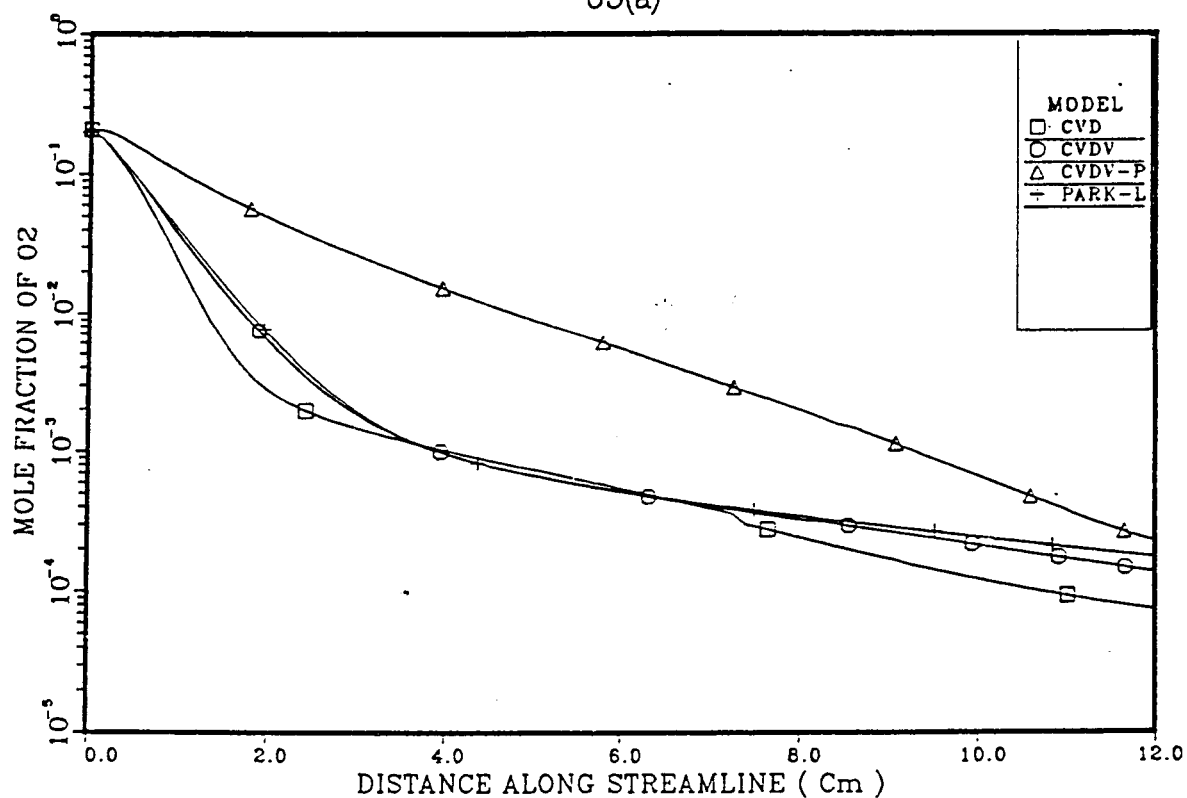


64(b)

FIGURES 64(a),64(b).PROFILES AT V=7.7 Km/s, RR2

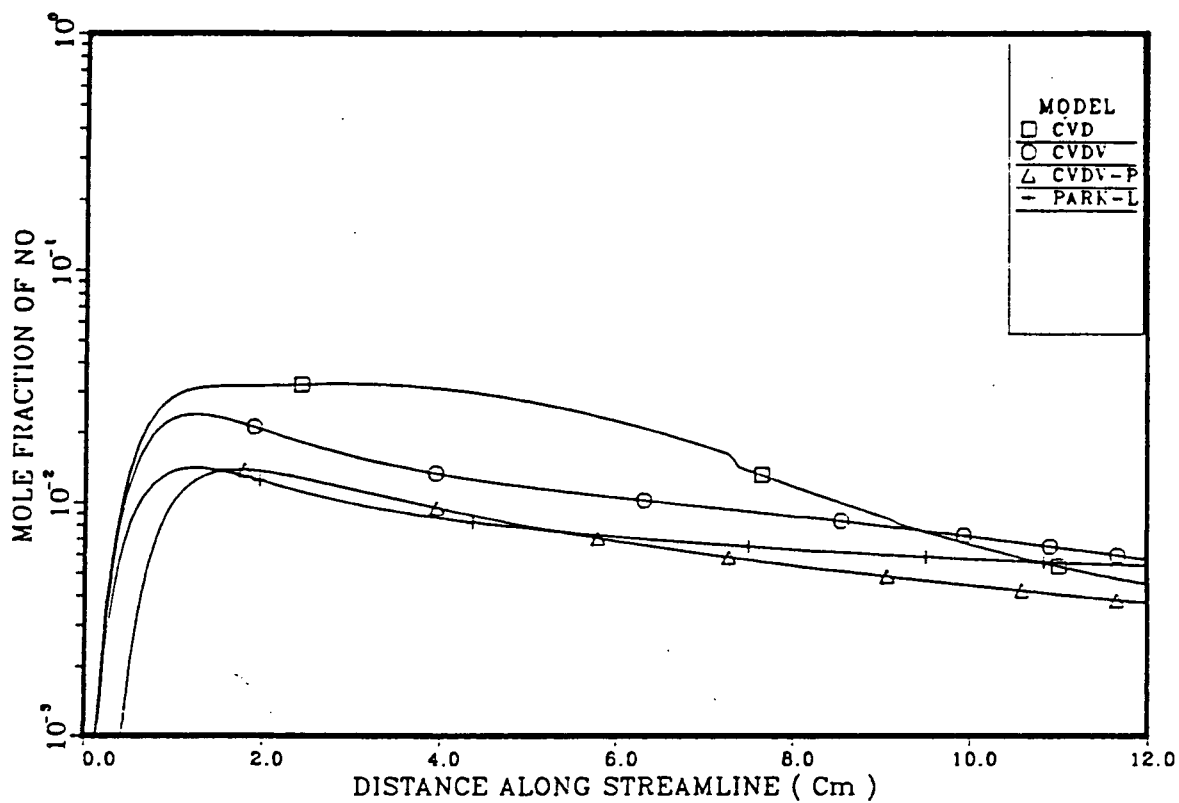


65(a)

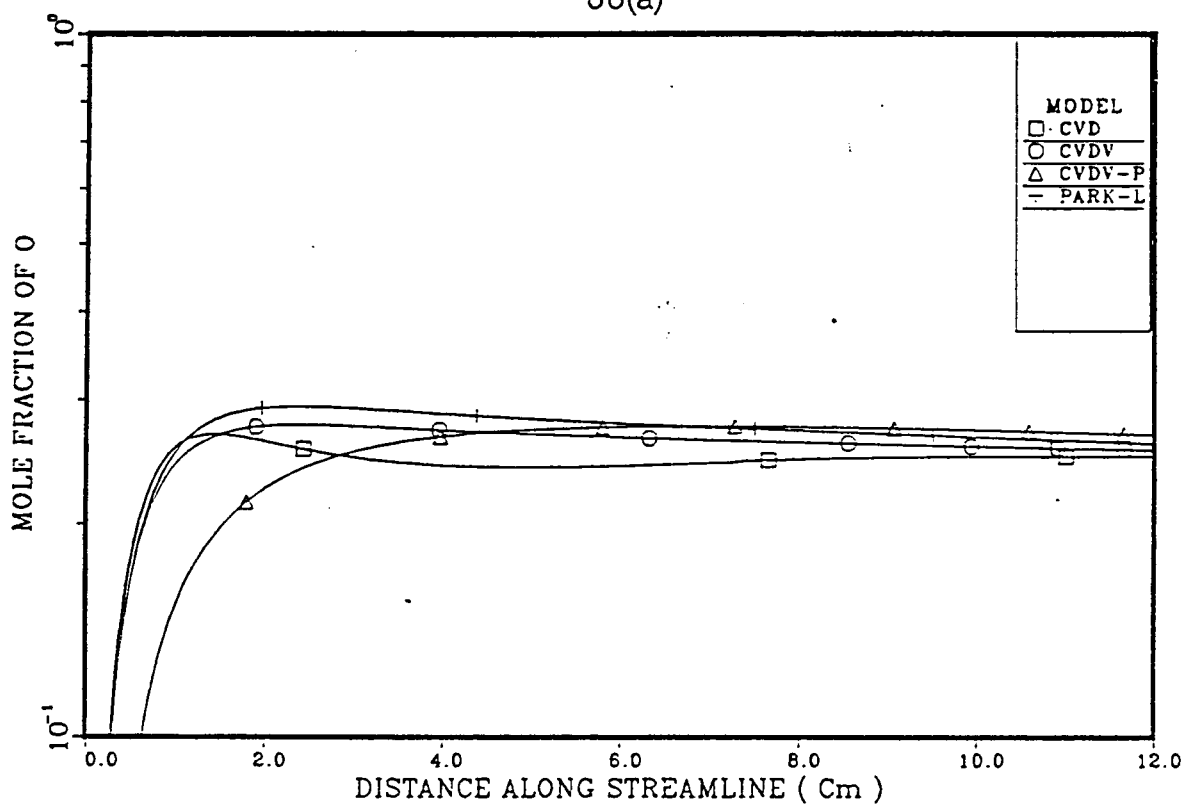


65(b)

FIGURES 65(a),65(b).PROFILES AT V=7.7 Km/s, RR2

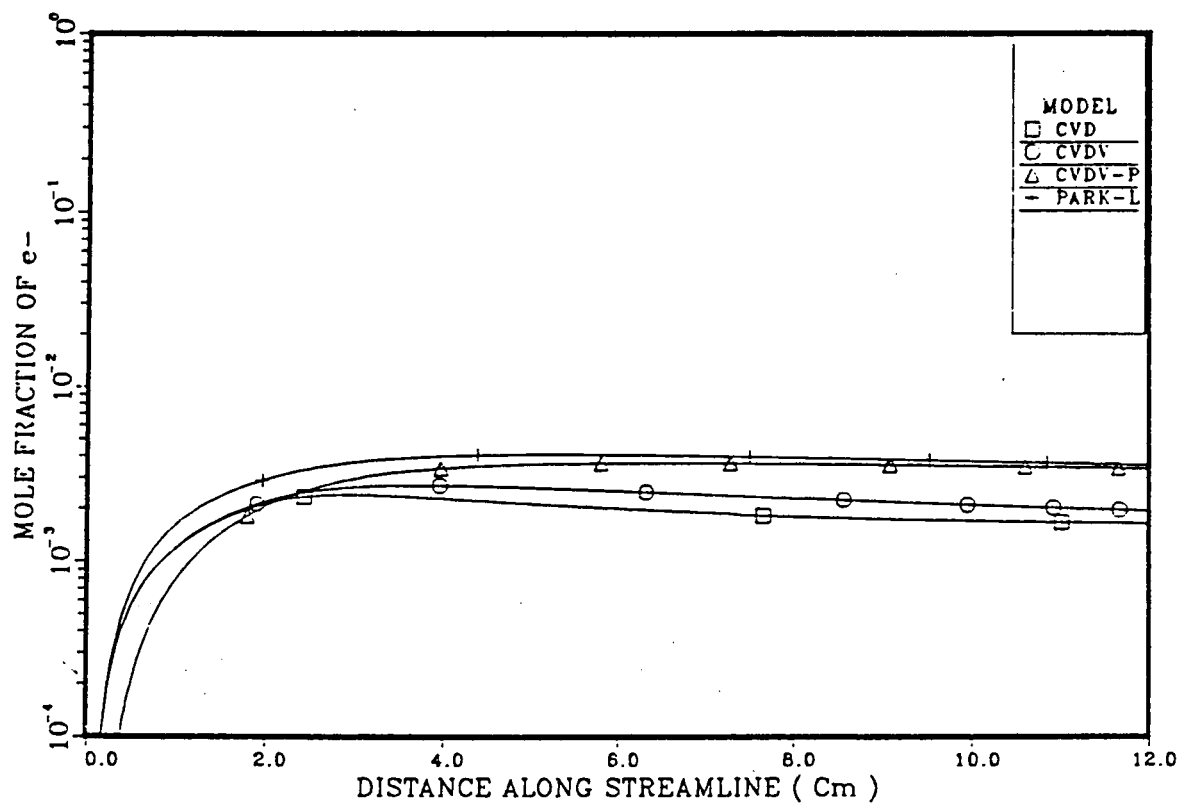


66(a)

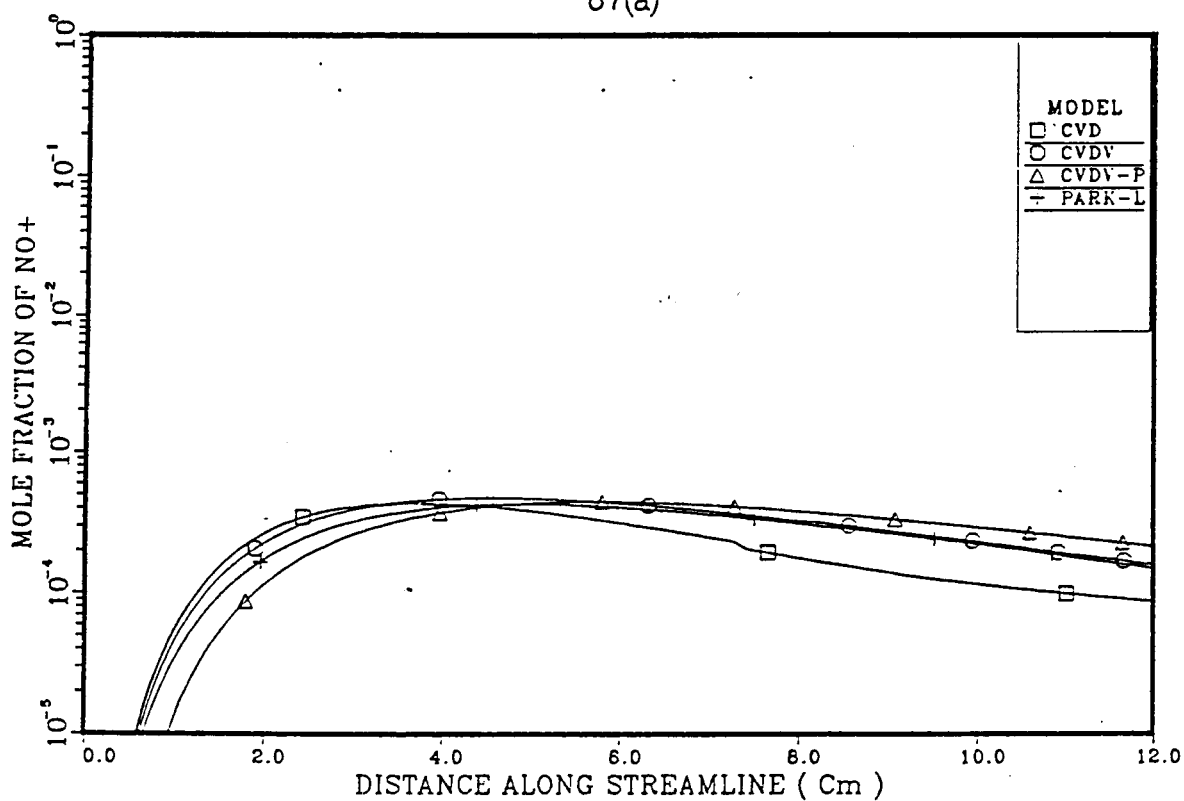


66(b)

FIGURES 66(a),66(b).PROFILES AT V=7.7 Km/s, RR2

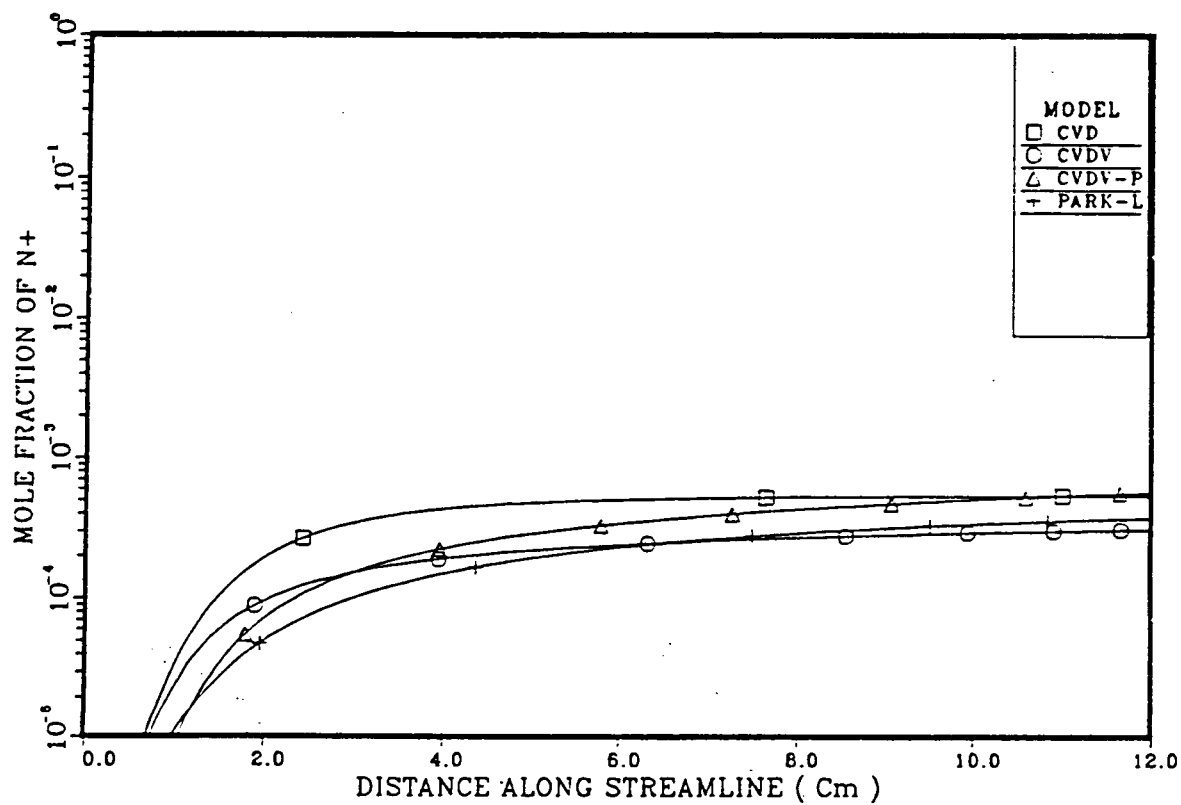


67(a)

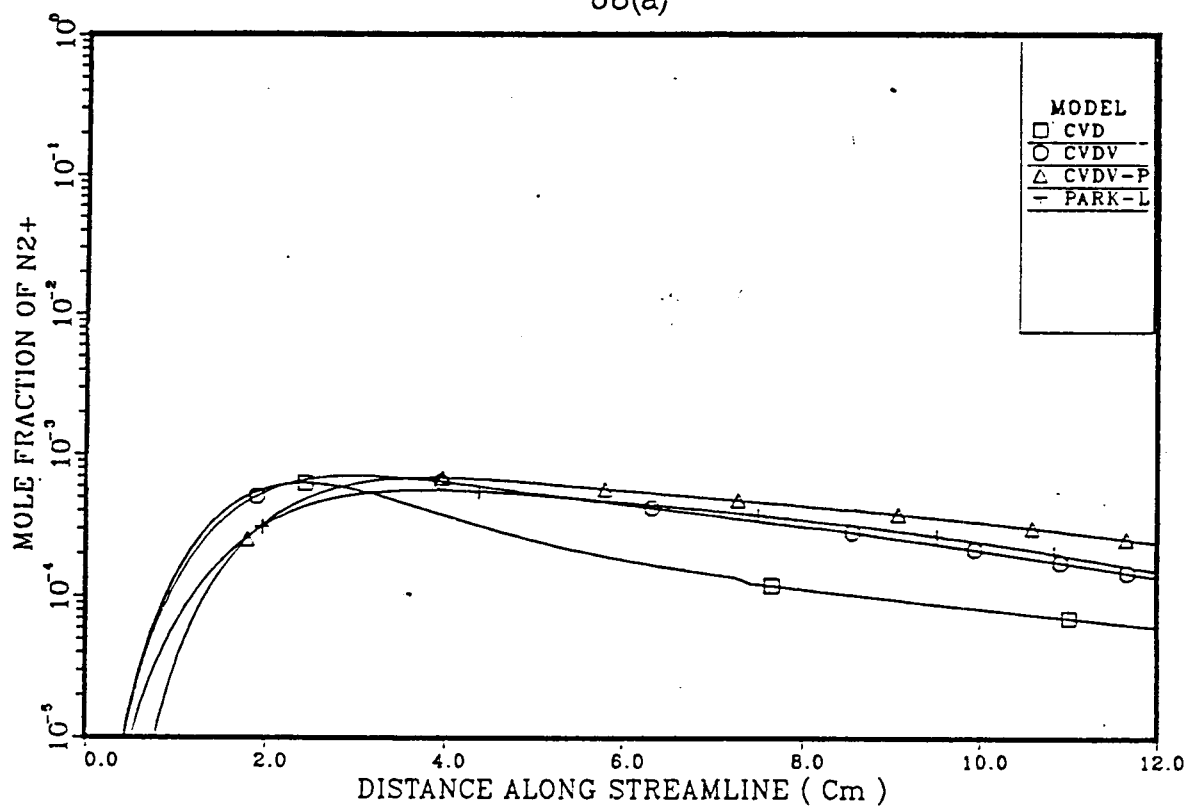


67(b)

FIGURES 67(a),67(b).PROFILES AT V=7.7 Km/s, RR2



68(a)



68(b)

FIGURES 68(a),68(b).PROFILES AT $V=7.7$ Km/s, RR2

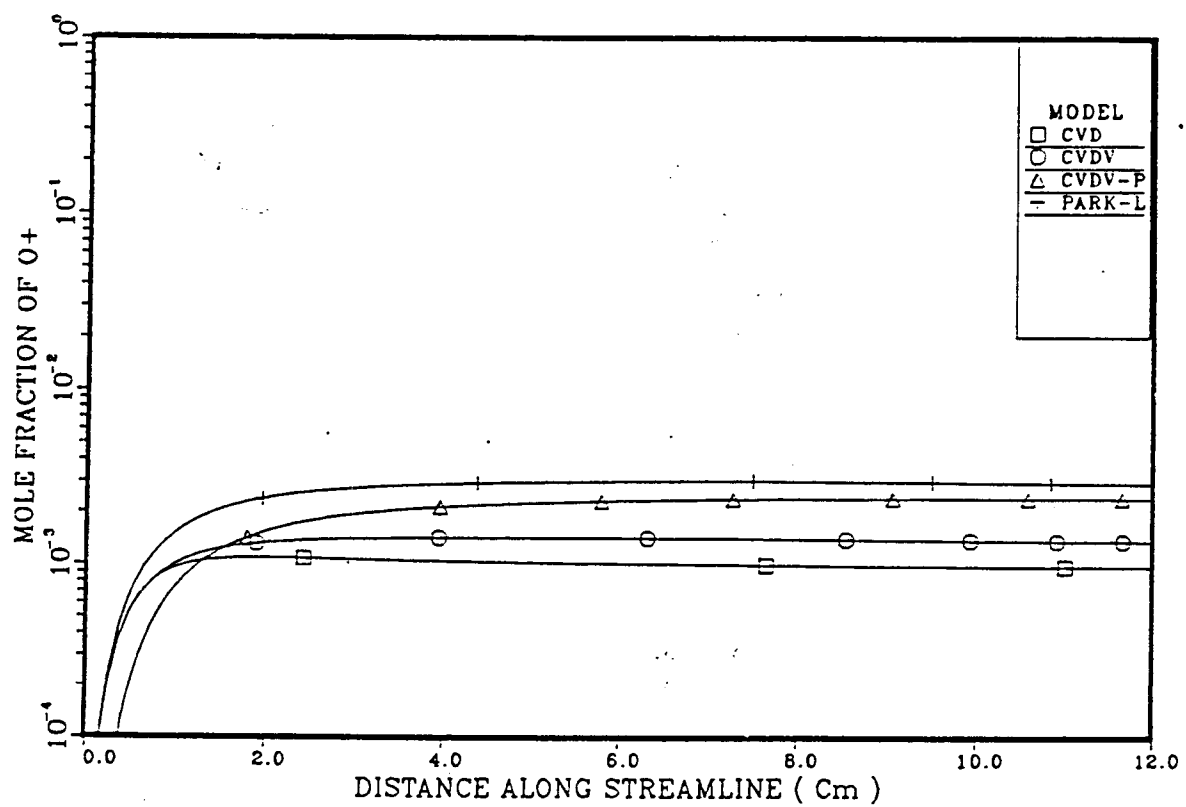
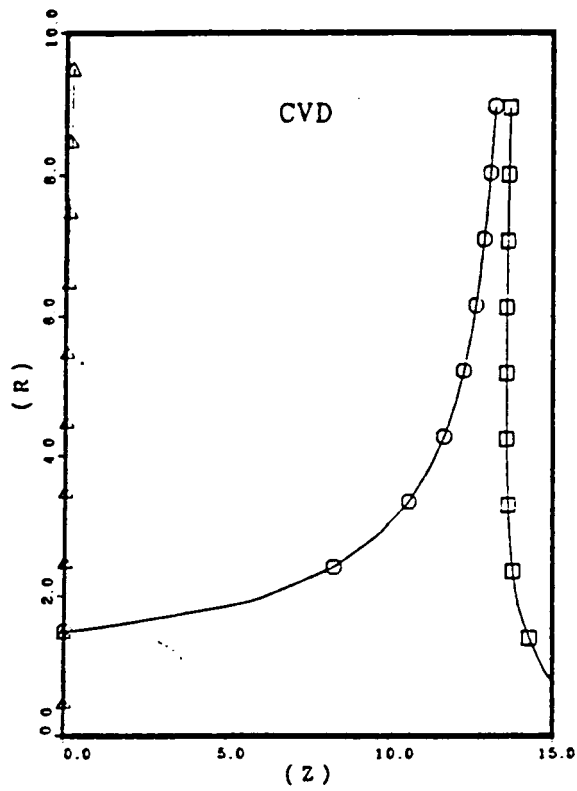
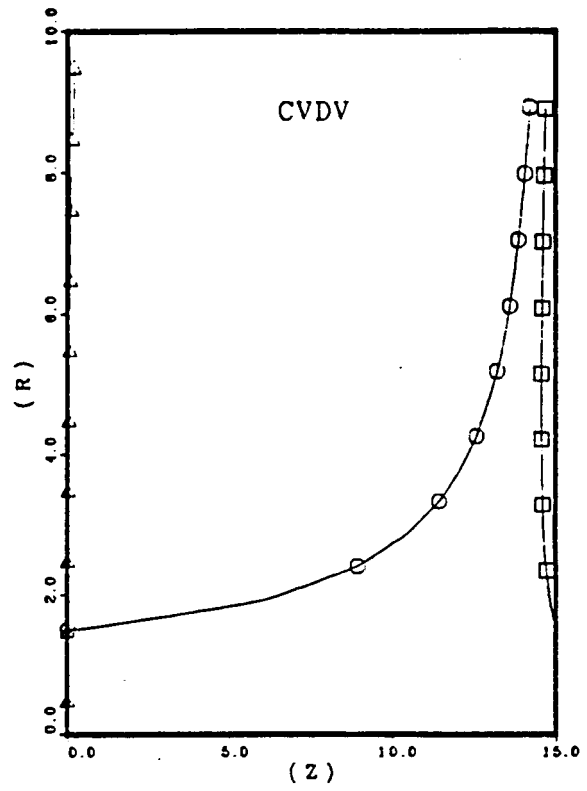


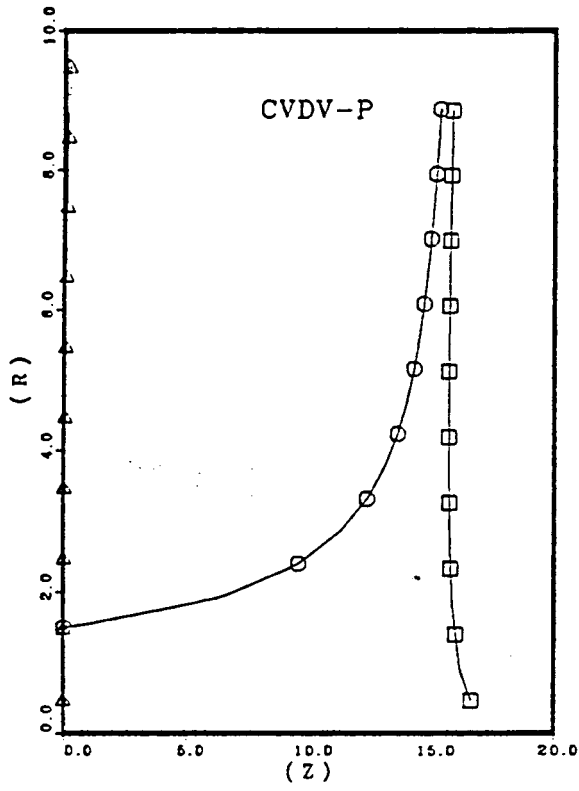
FIGURE 69.PROFILE AT V=7.7 Km/s, RR2



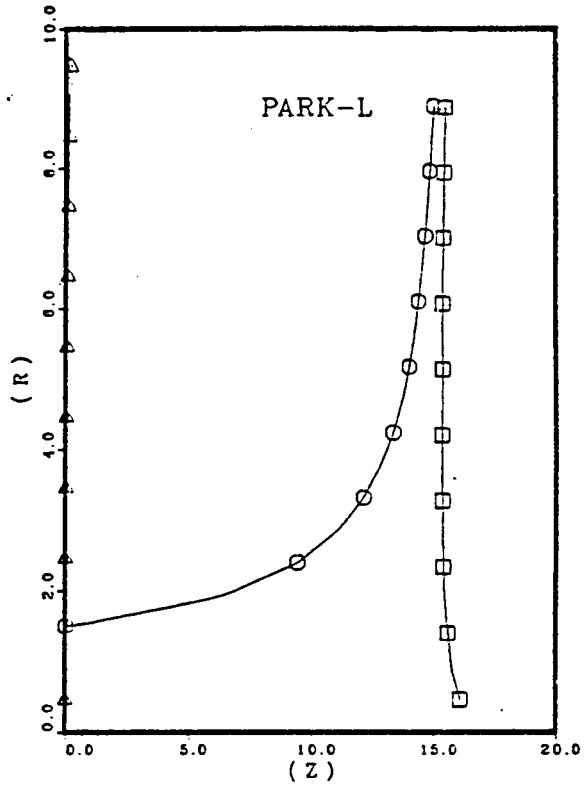
70(a)



70(b)

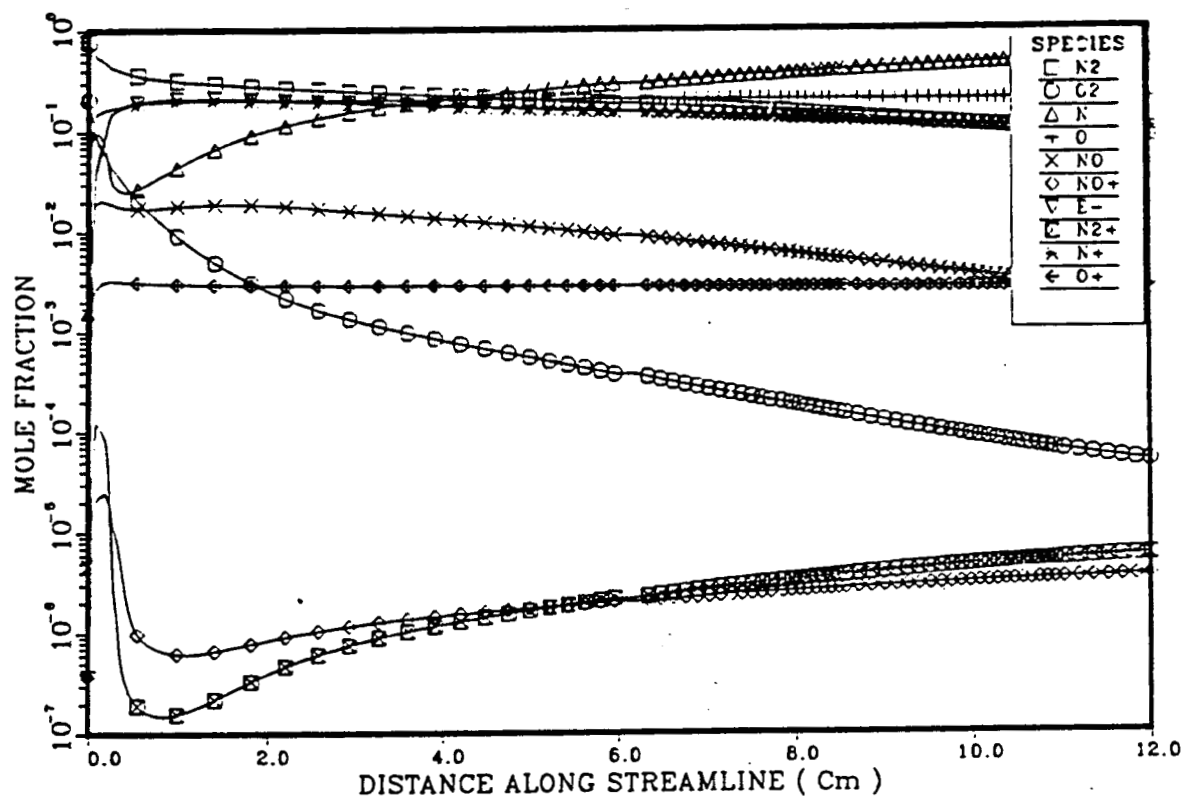


70(c)

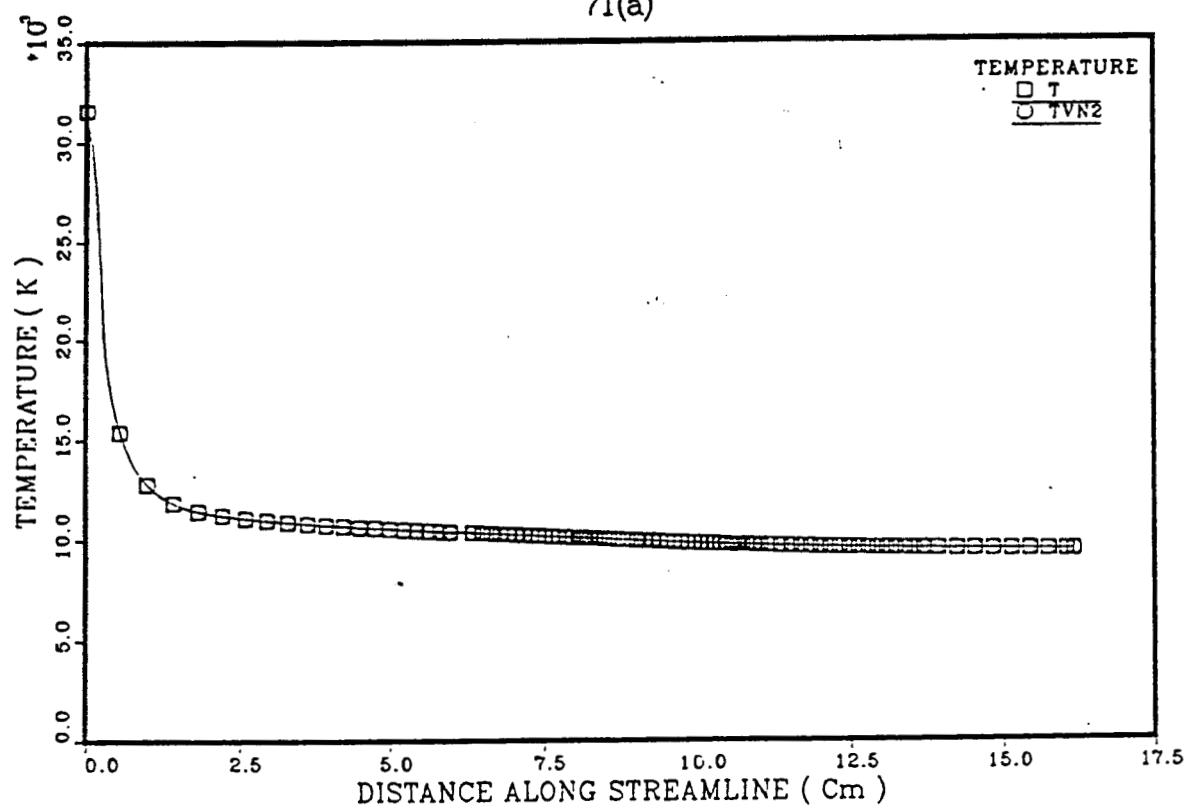


70(d)

FIGURES 70(a),70(b),70(c),70(d).COORD,V=7.7 Km/s, RR2

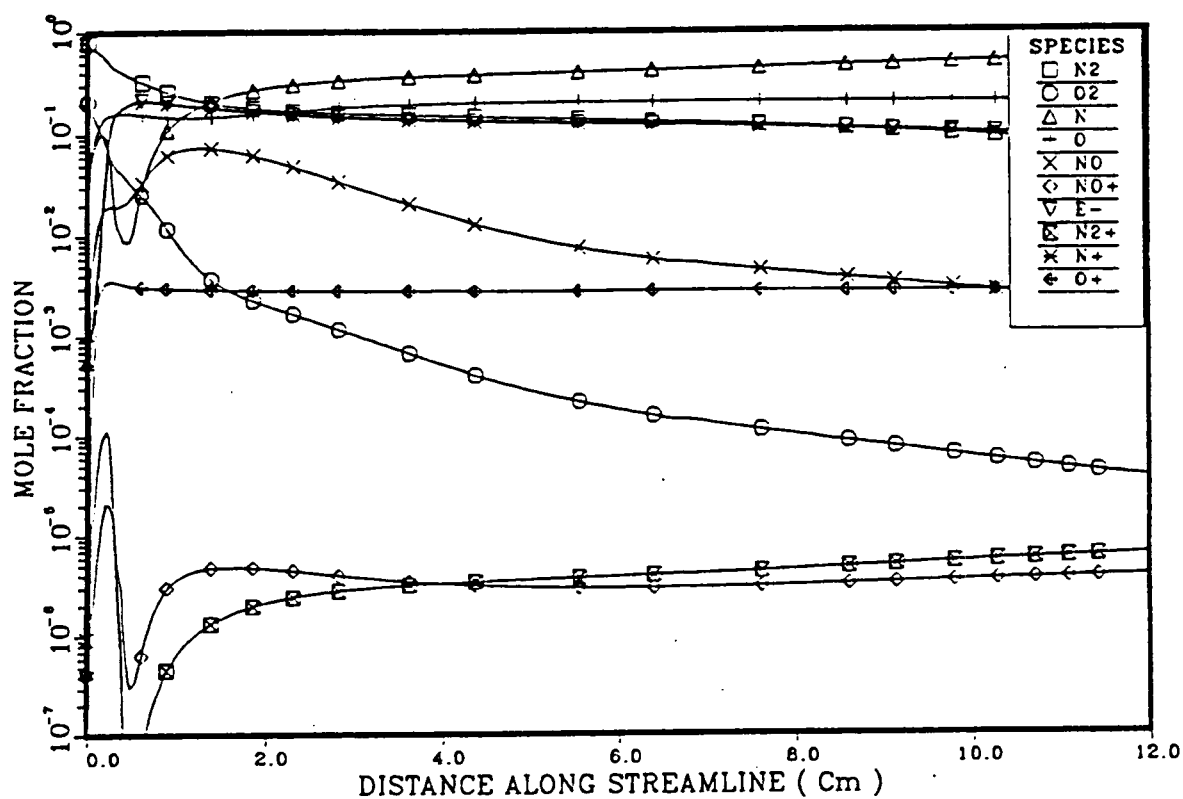


71(a)

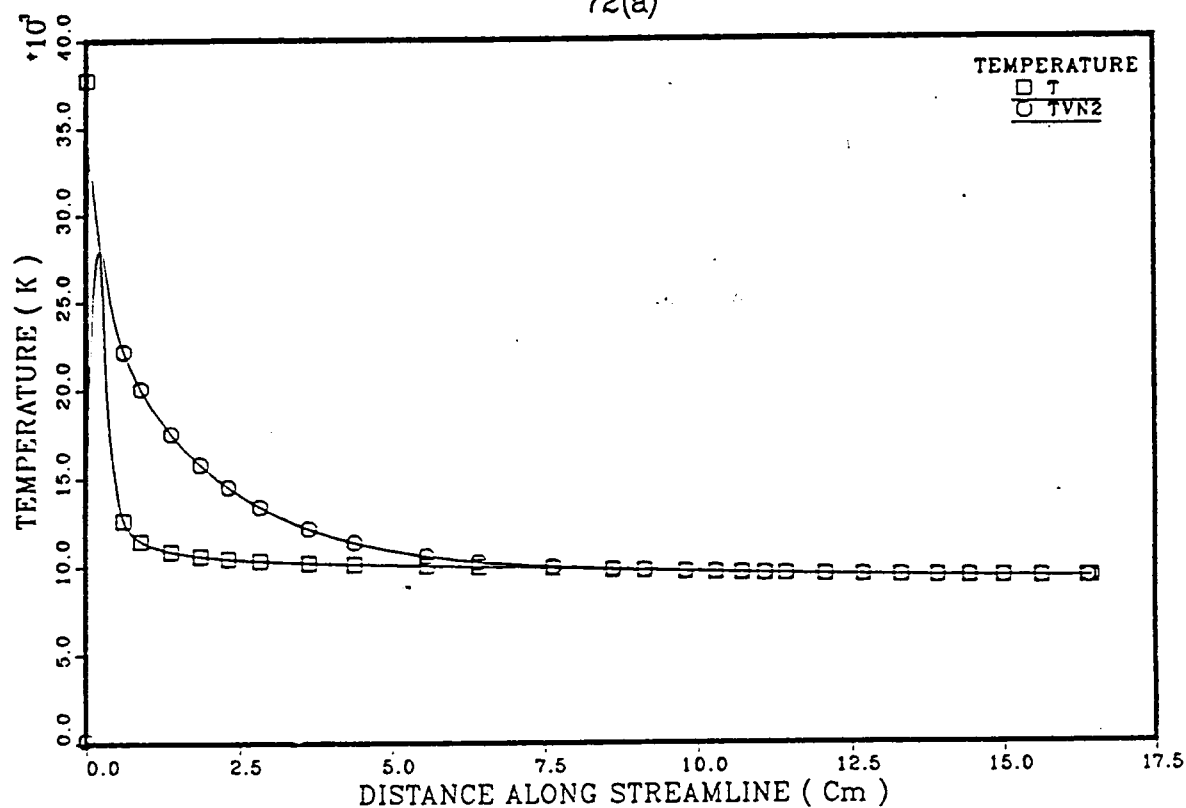


71(b)

FIGURES 71(a),71(b).VEQ MODEL AT V=10 Km/s, RR3

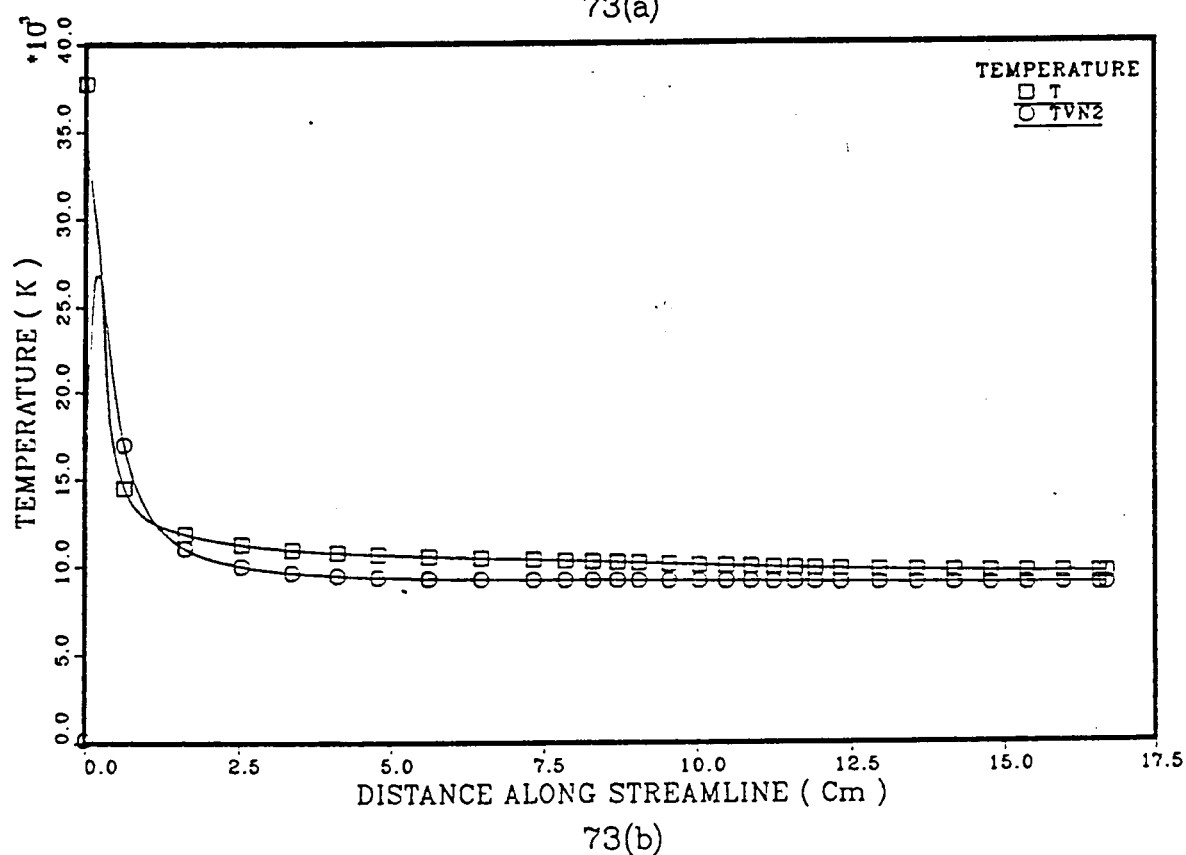
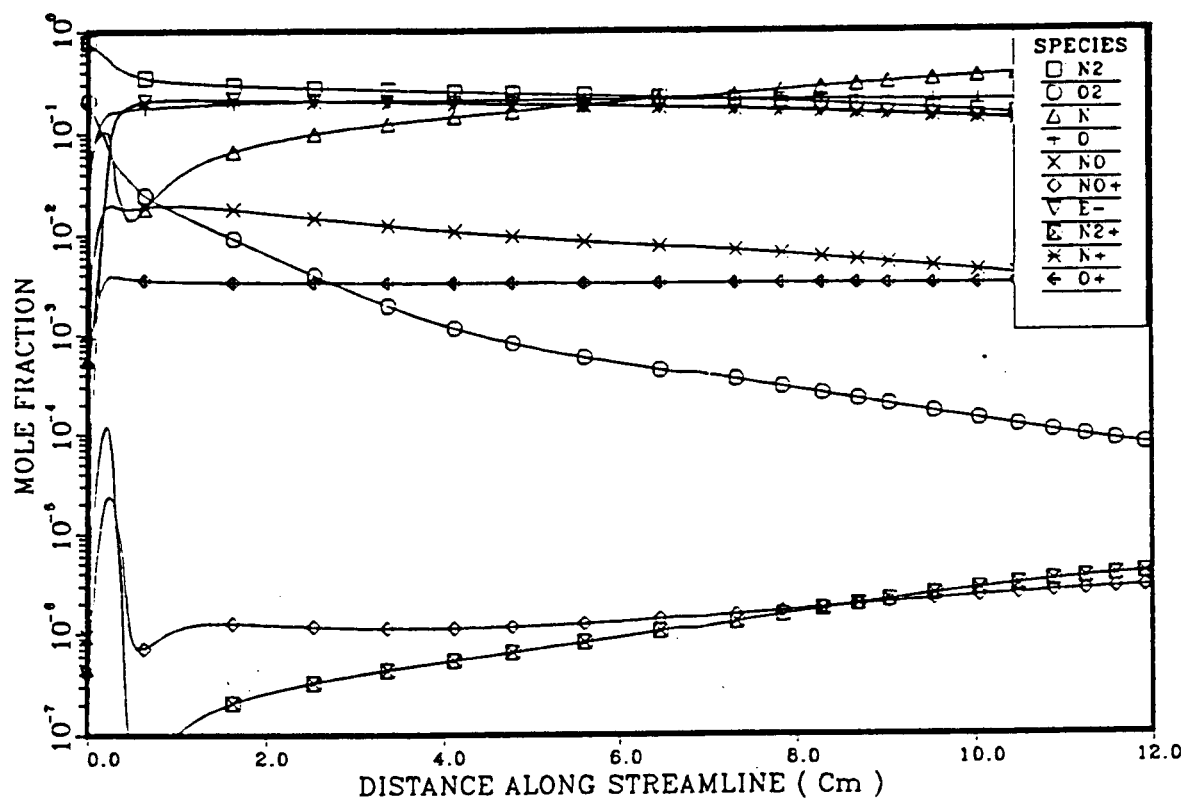


72(a)

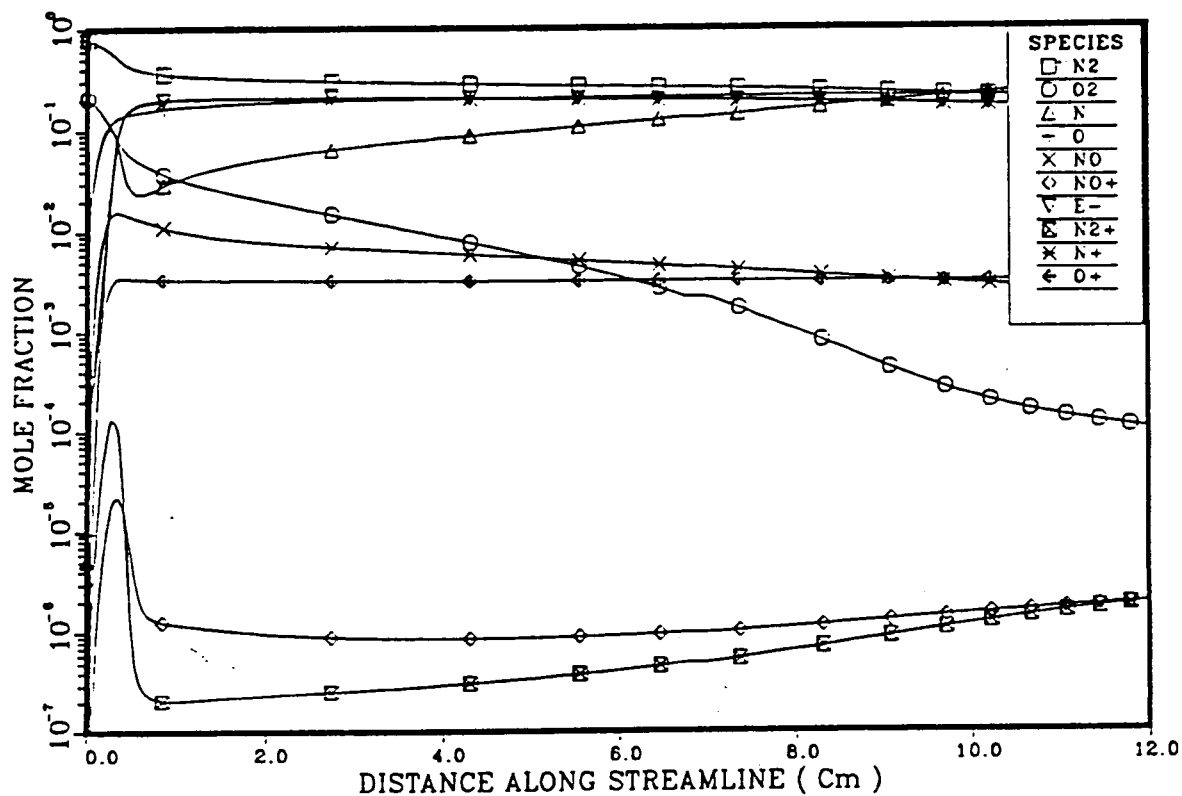


72(b)

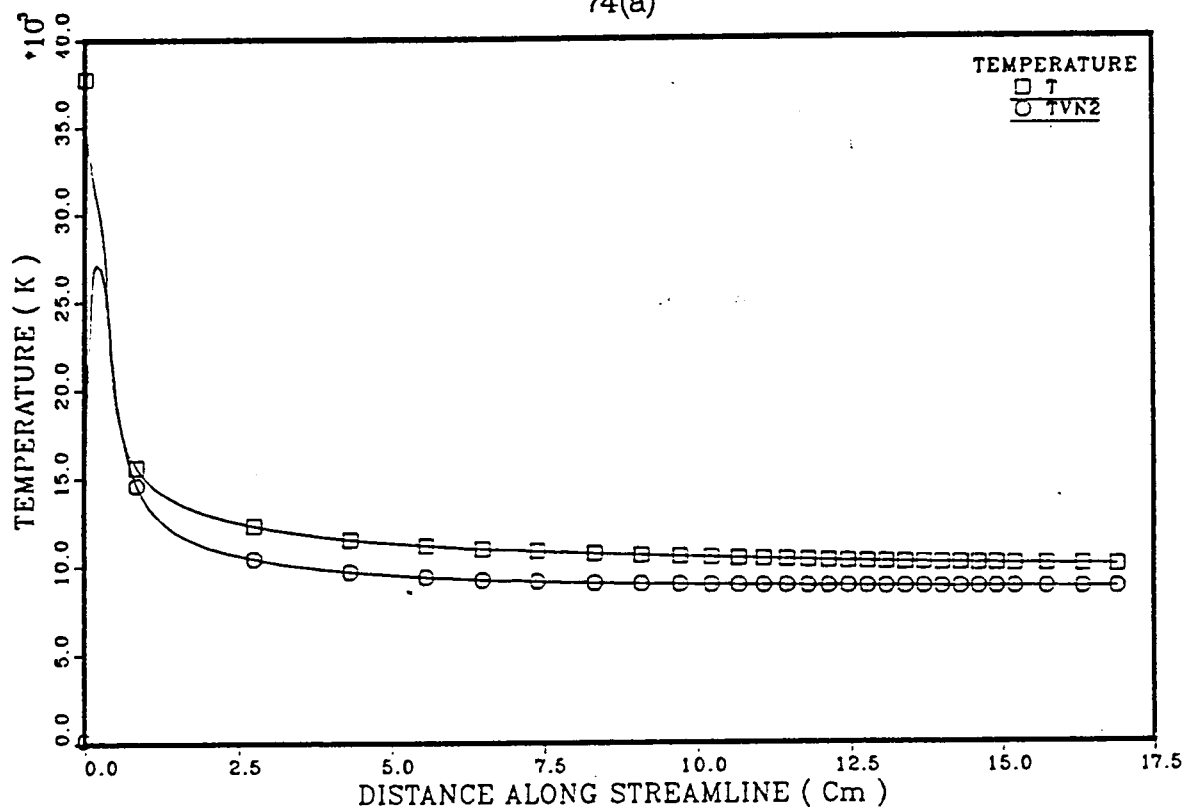
FIGURES 72(a),72(b).CVD MODEL AT $V=10$ Km/s, RR3



FIGURES 73(a),73(b).CVDV MODEL AT V=10 Km/s, RR3

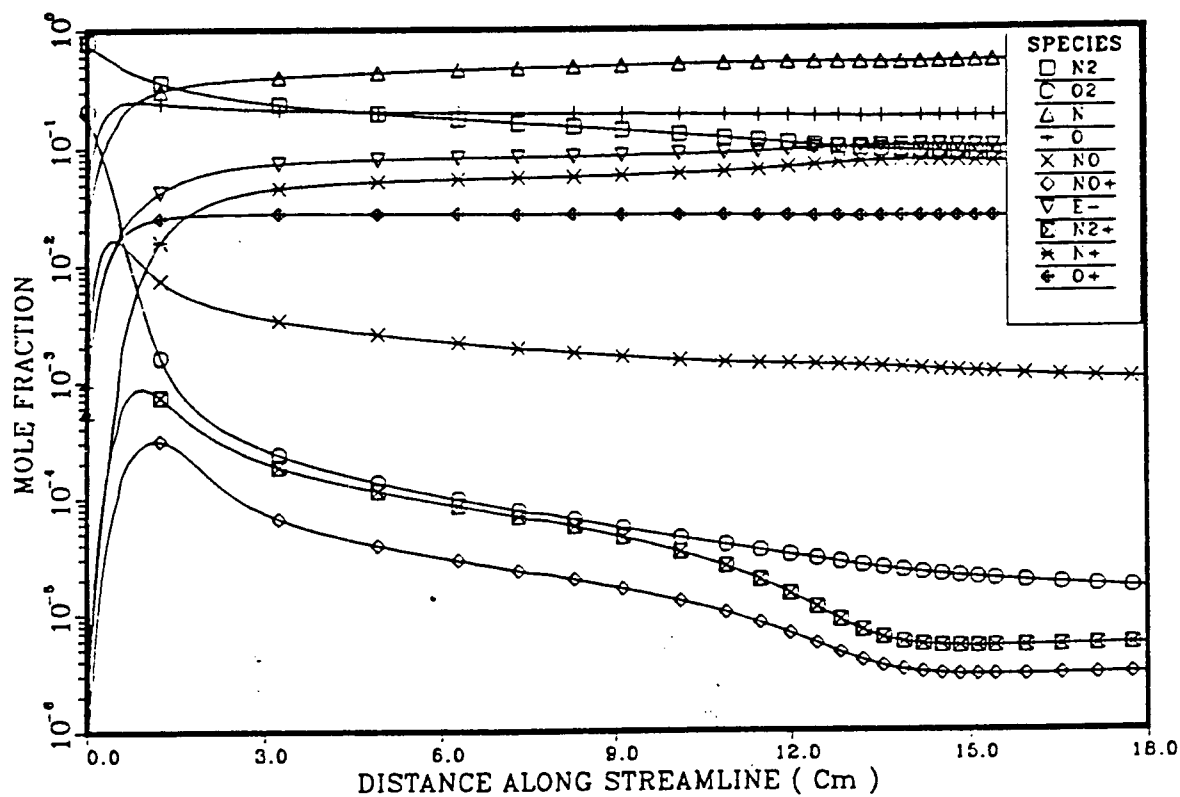


74(a)

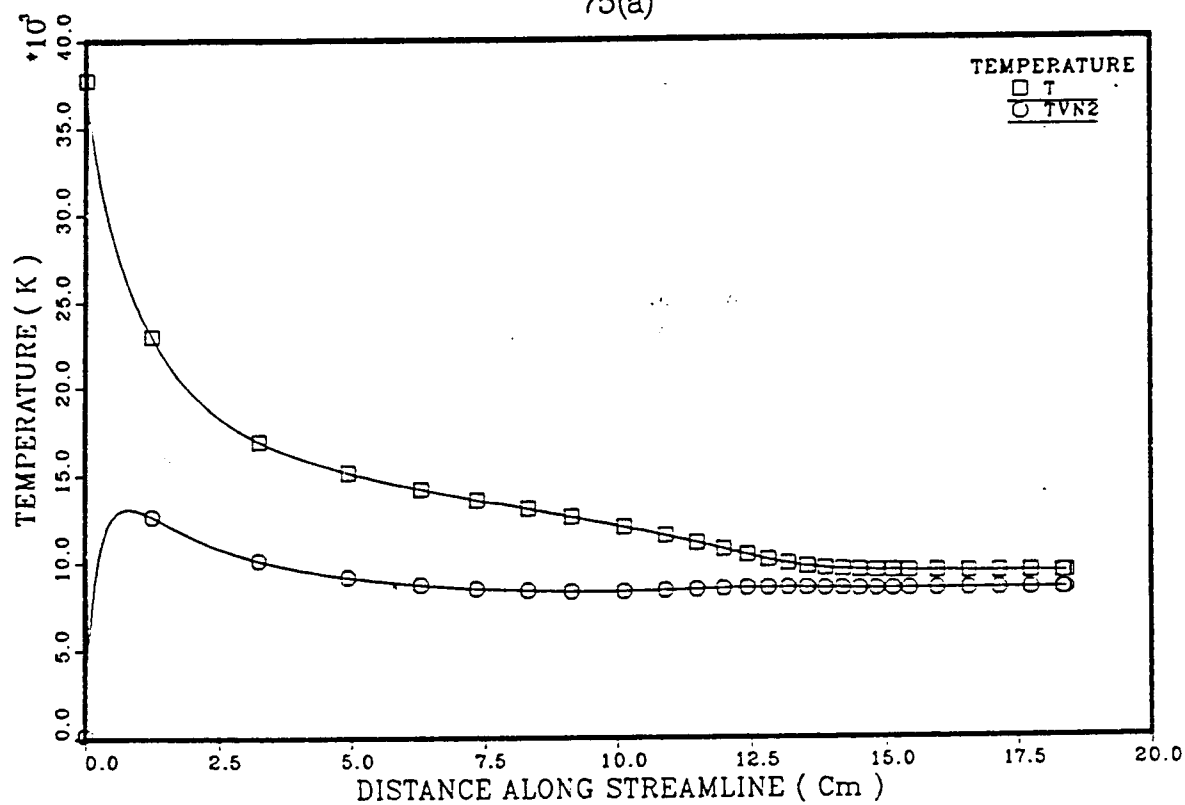


74(b)

FIGURES 74(a),74(b).CVDV-P MODEL AT V=10 Km/s, RR3

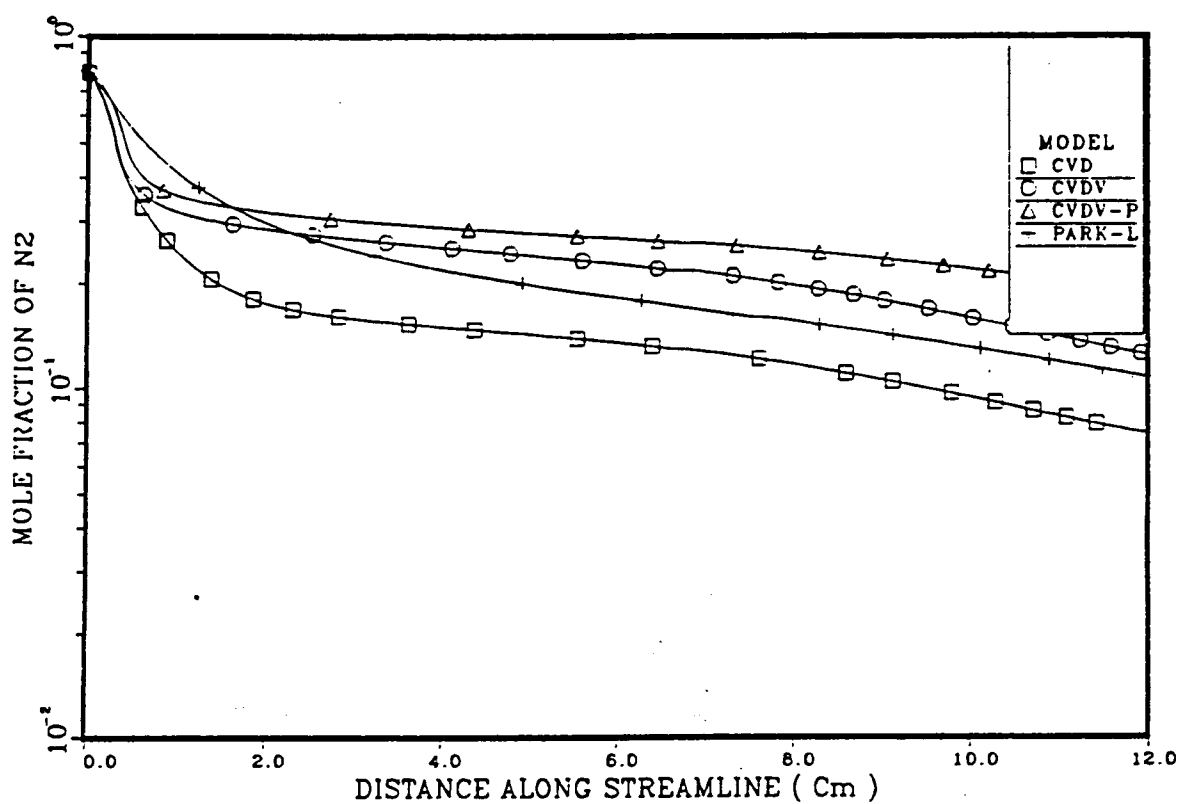


75(a)

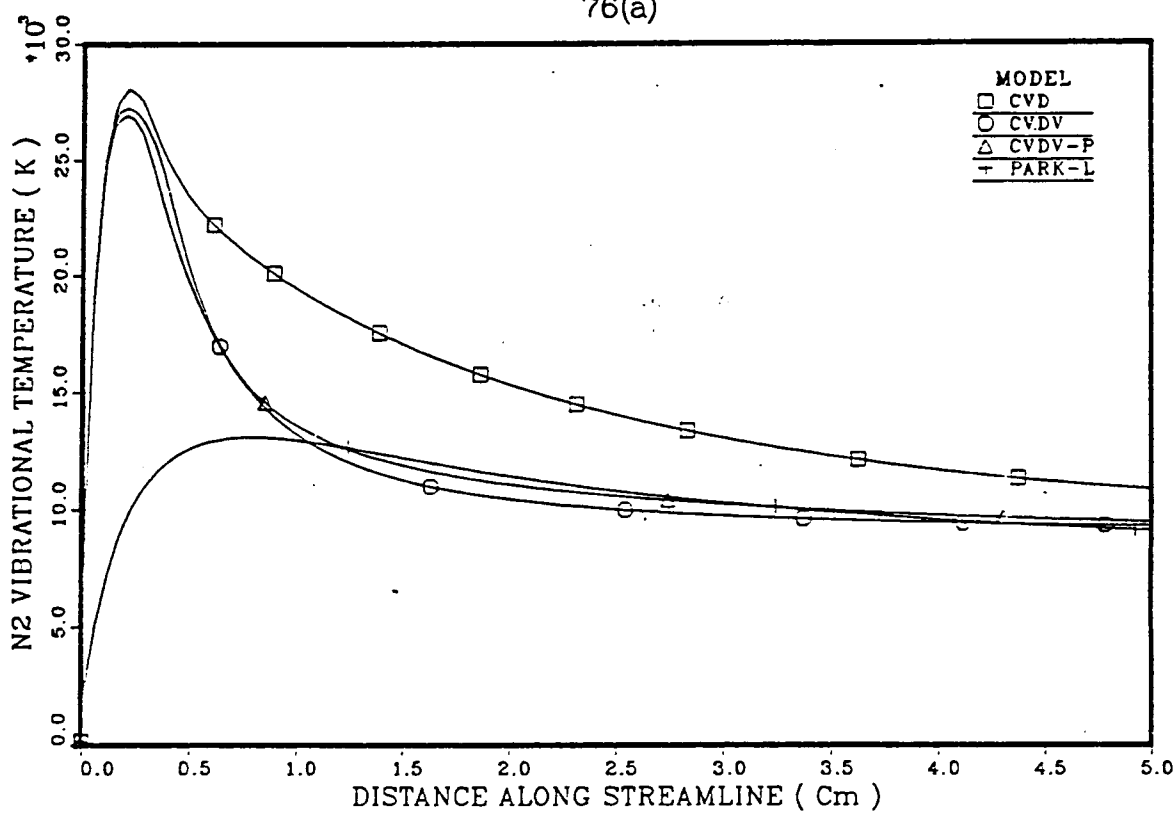


75(b)

FIGURES 75(a),75(b).PARK-L MODEL AT V=10 Km/s, RR3

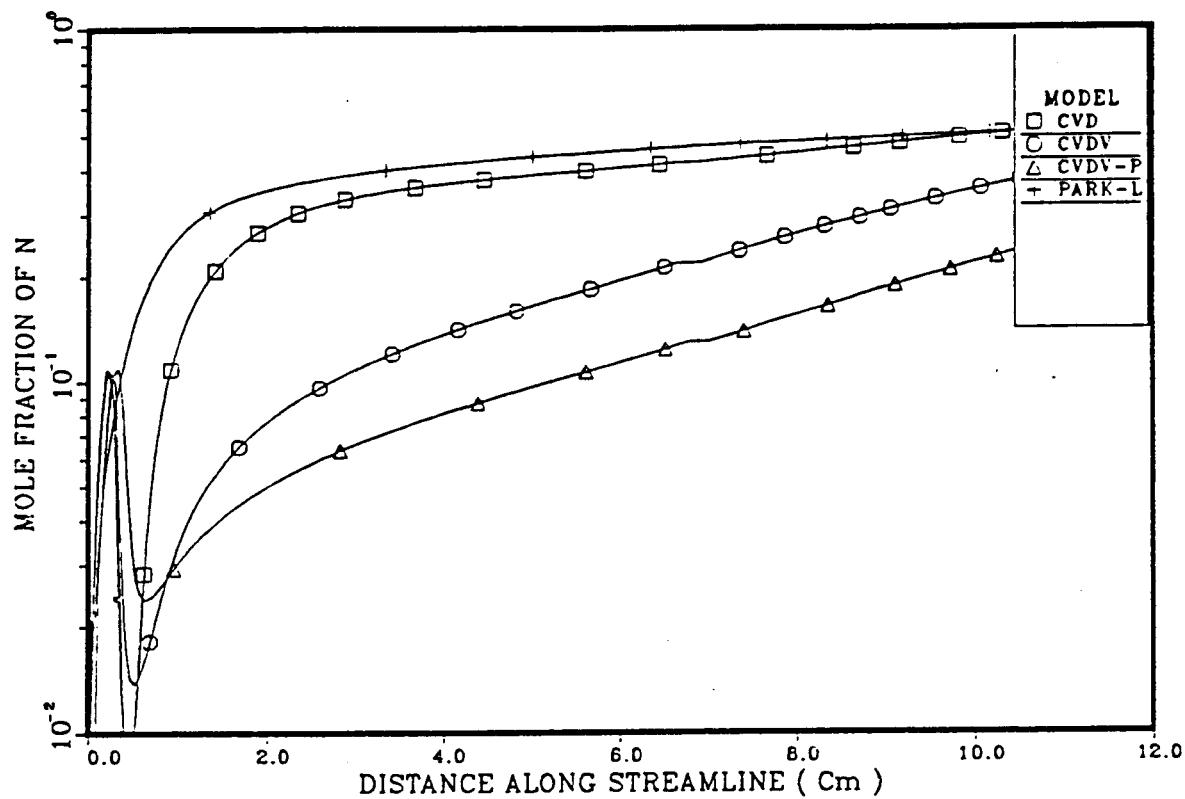


76(a)

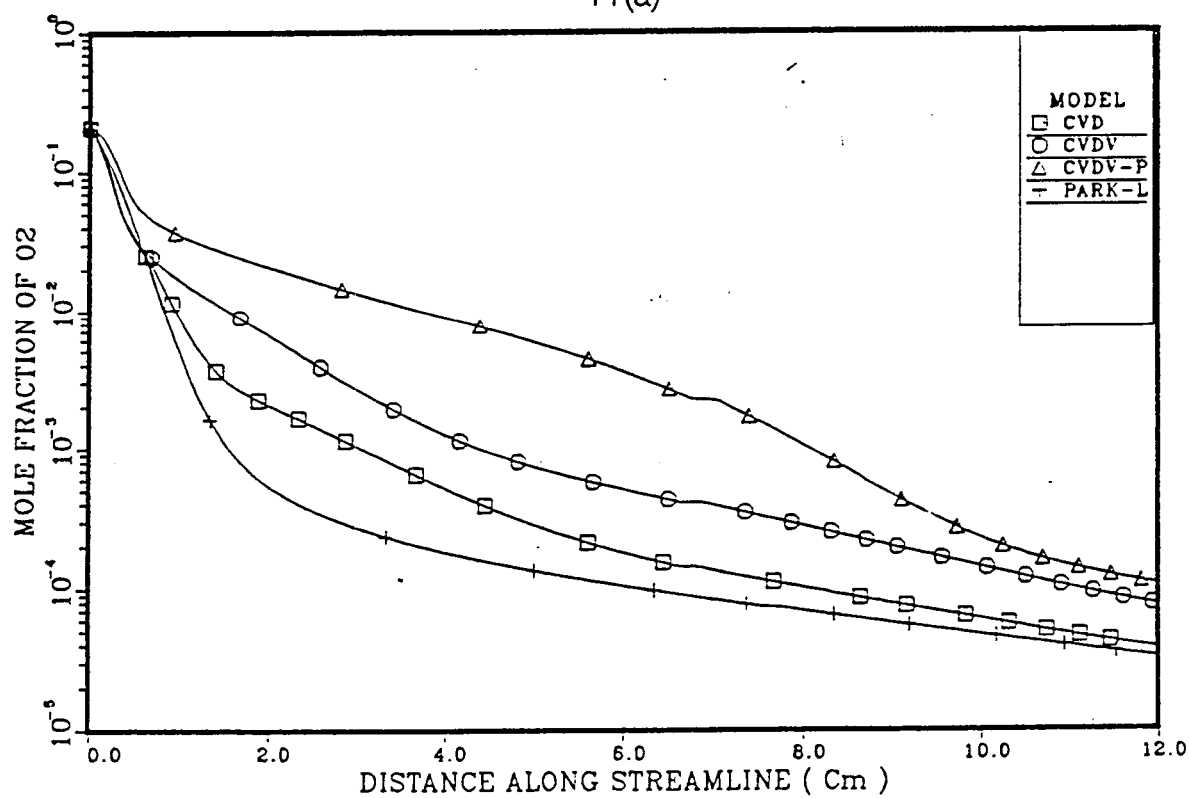


76(b)

FIGURES 76(a),76(b).PROFILES AT V=10 Km/s, RR3

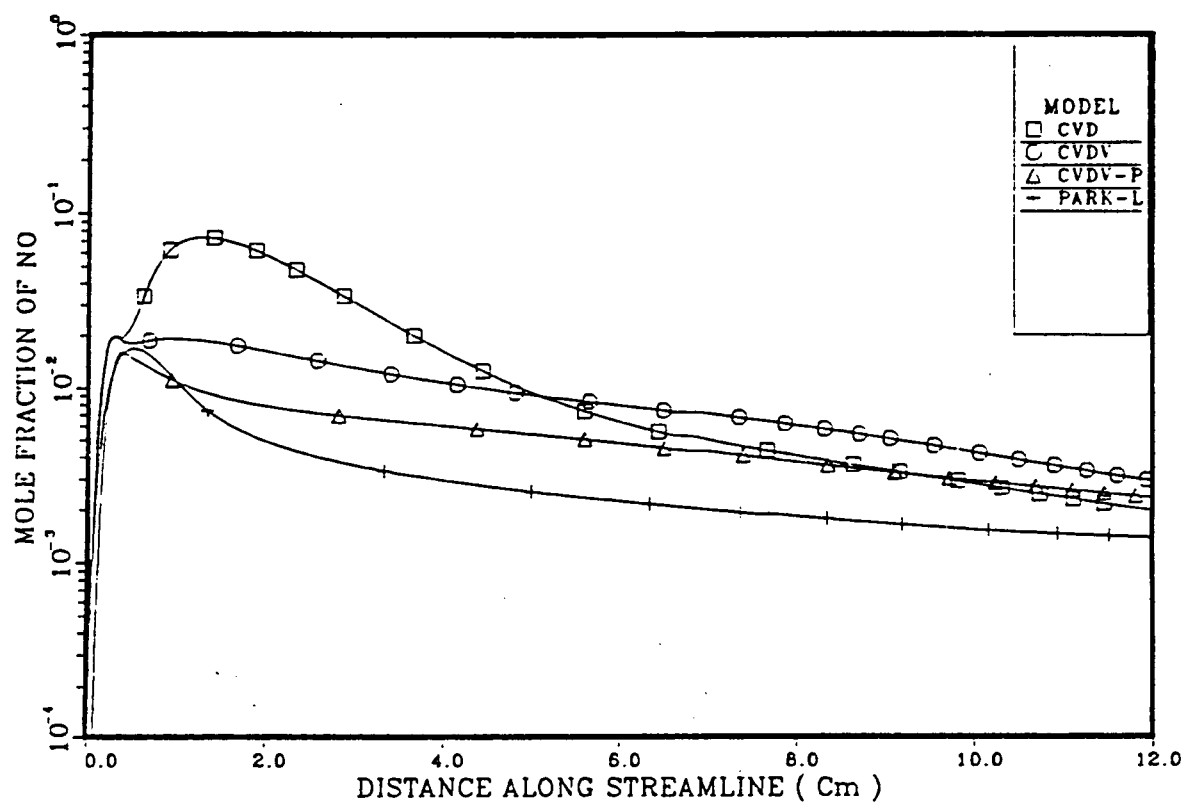


77(a)

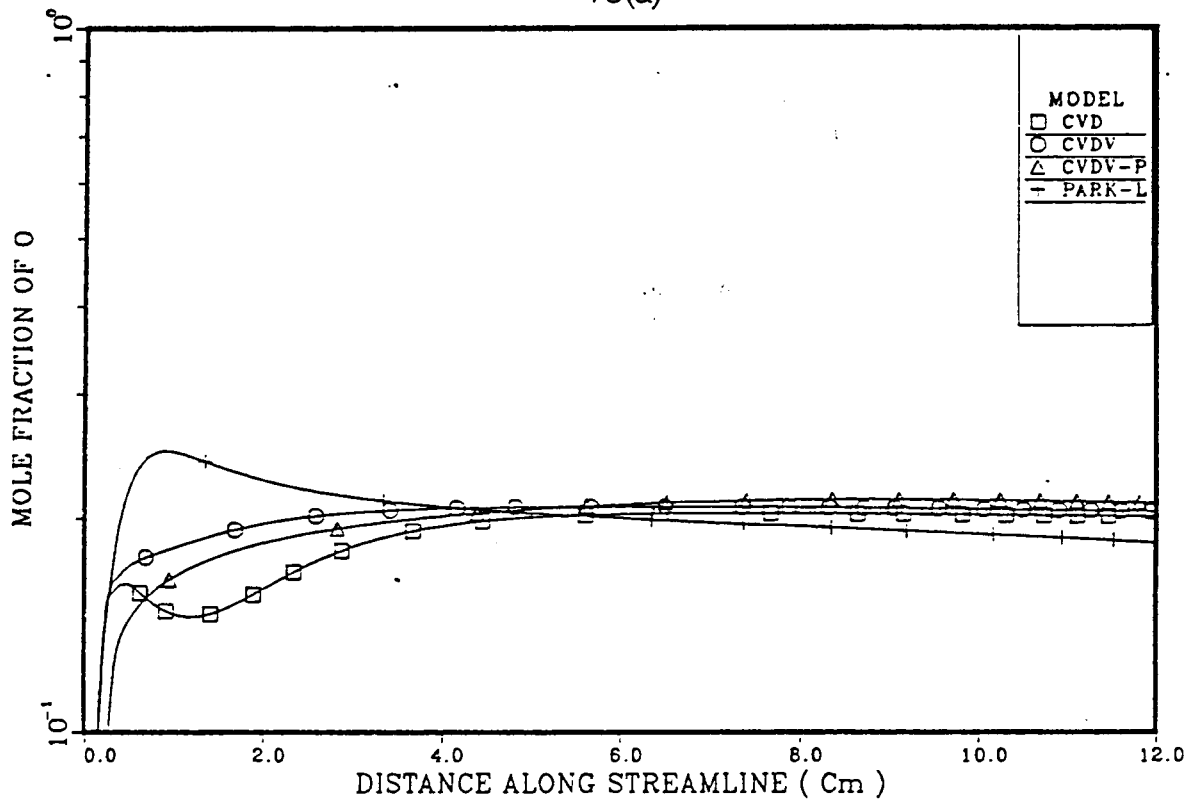


77(b)

FIGURES 77(a),77(b).PROFILES AT V=10 Km/s, RR3

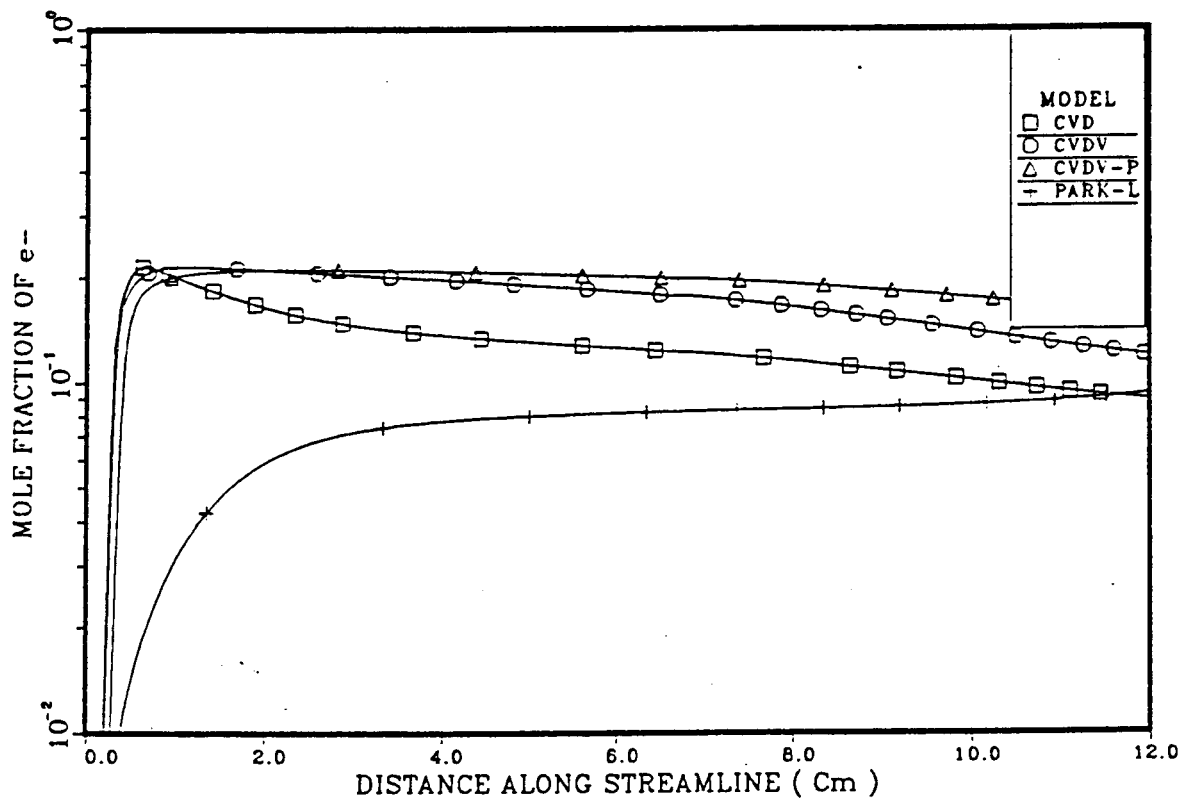


78(a)

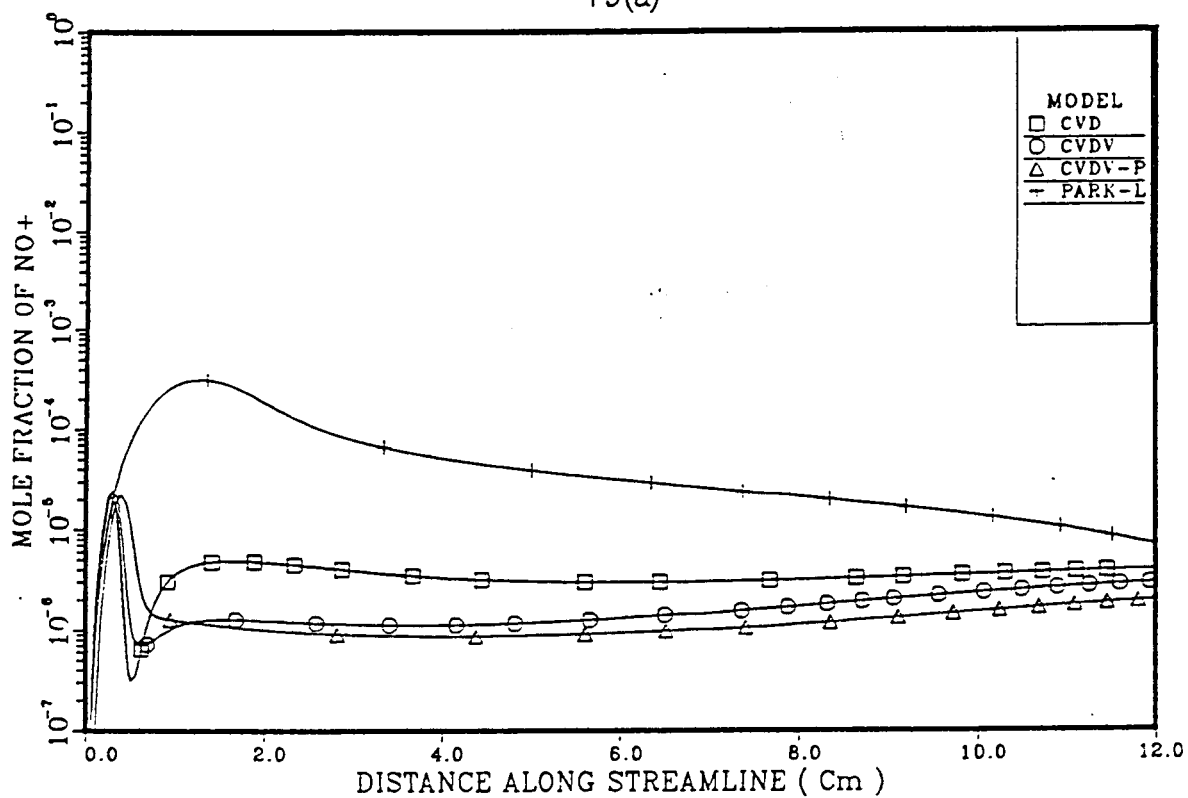


78(b)

FIGURES 78(a),78(b).PROFILES AT V=10 Km/s, RR3

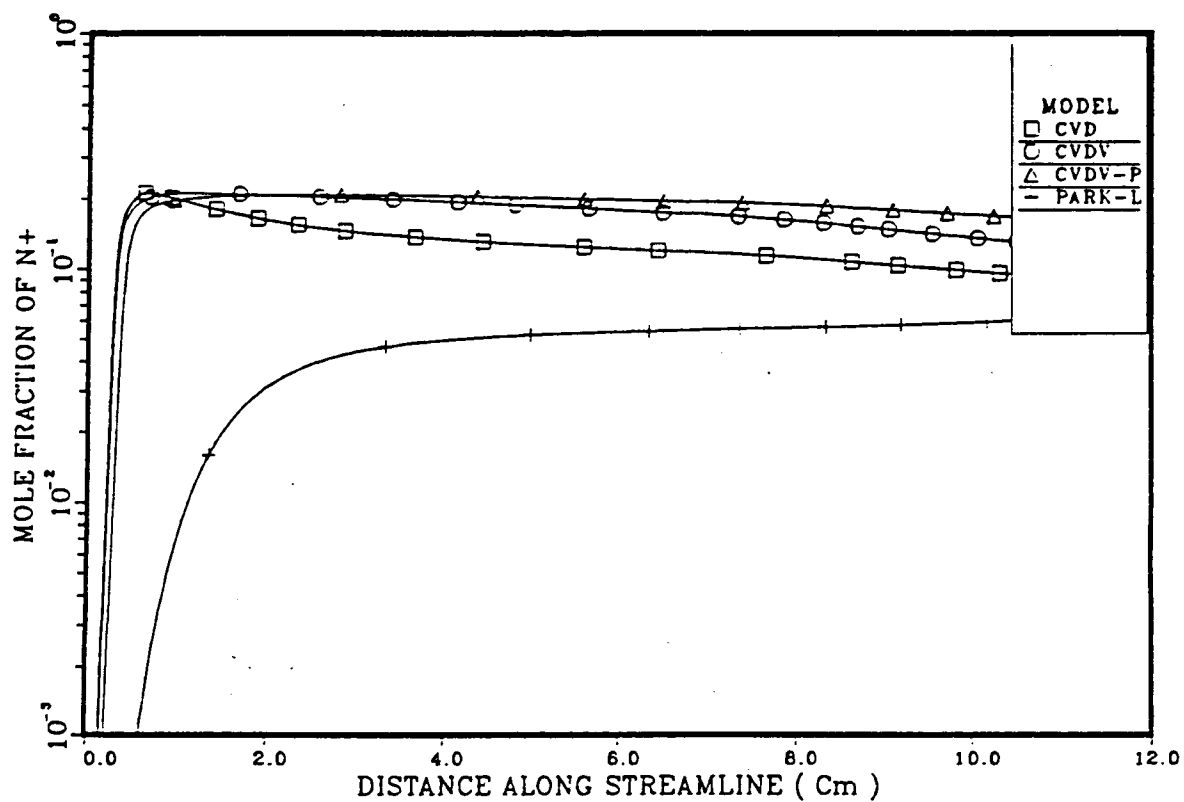


79(a)

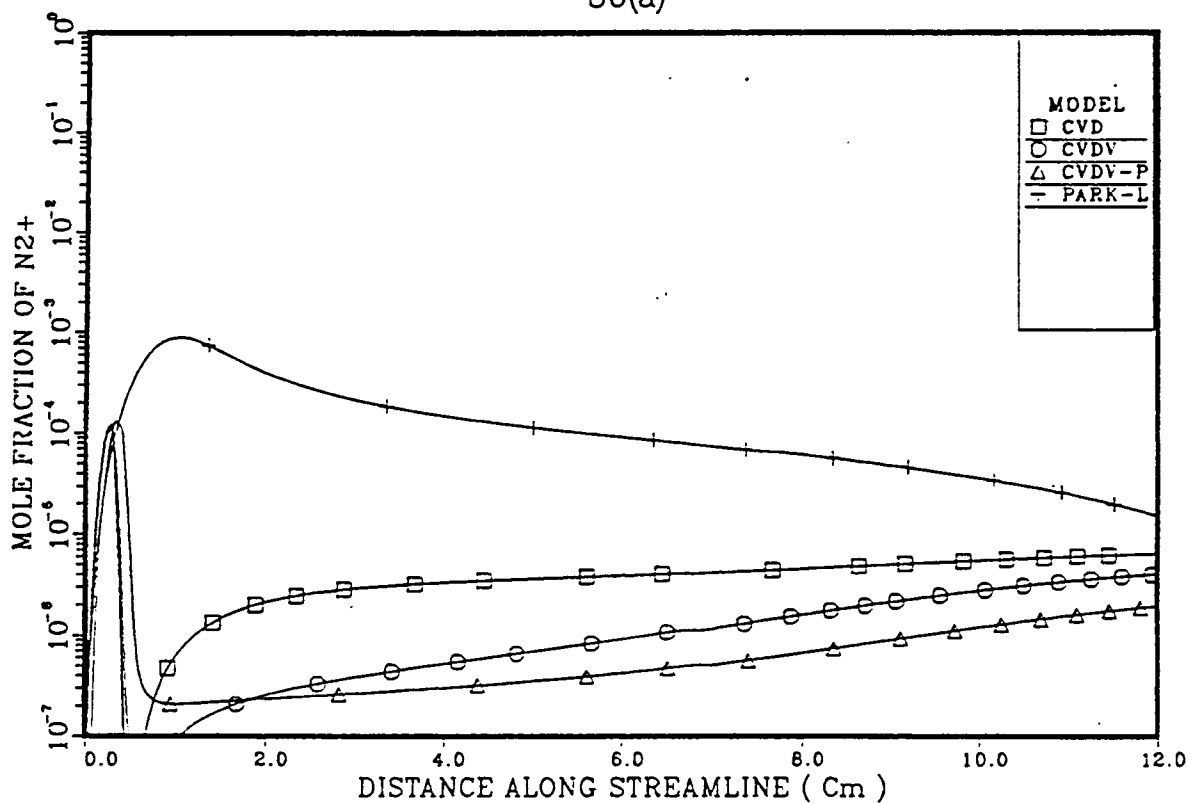


79(b)

FIGURES 79(a), 79(b). PROFILES AT $V=10$ Km/s, RR3



80(a)



80(b)

FIGURES 80(a),80(b).PROFILES AT $V=10$ Km/s, RR3

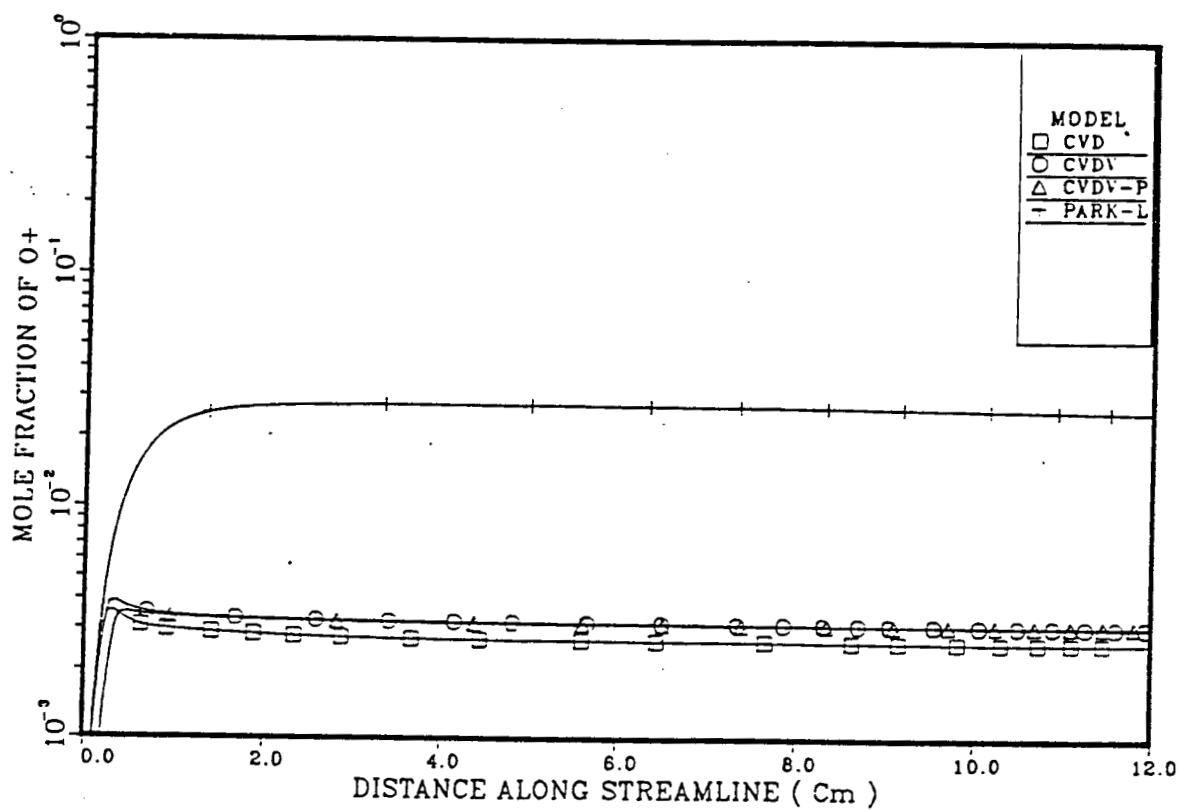
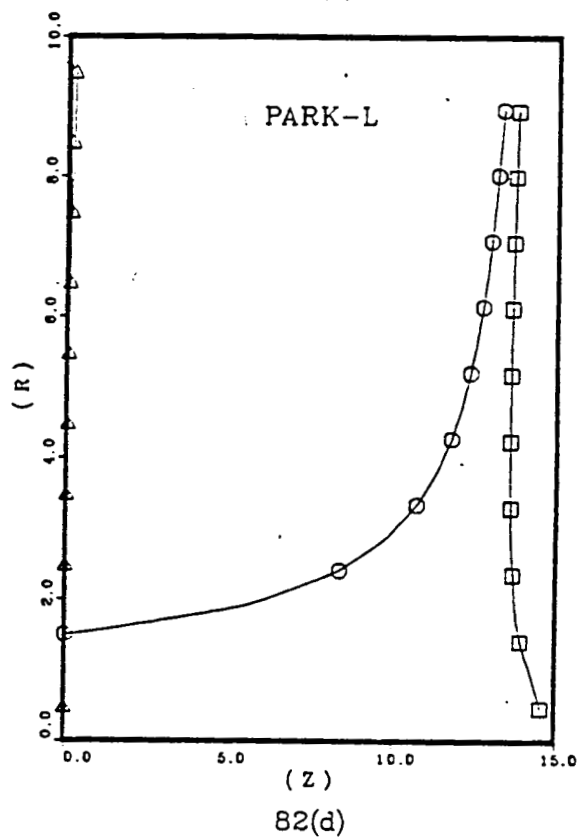
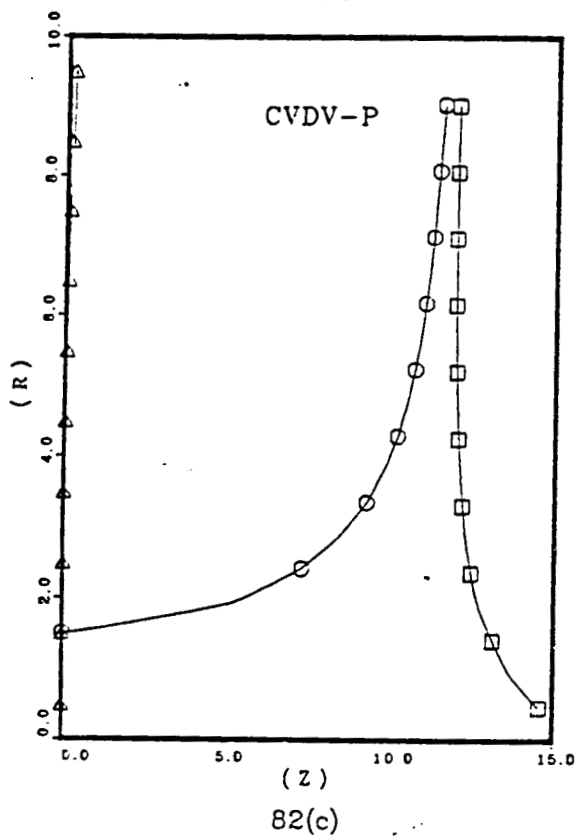
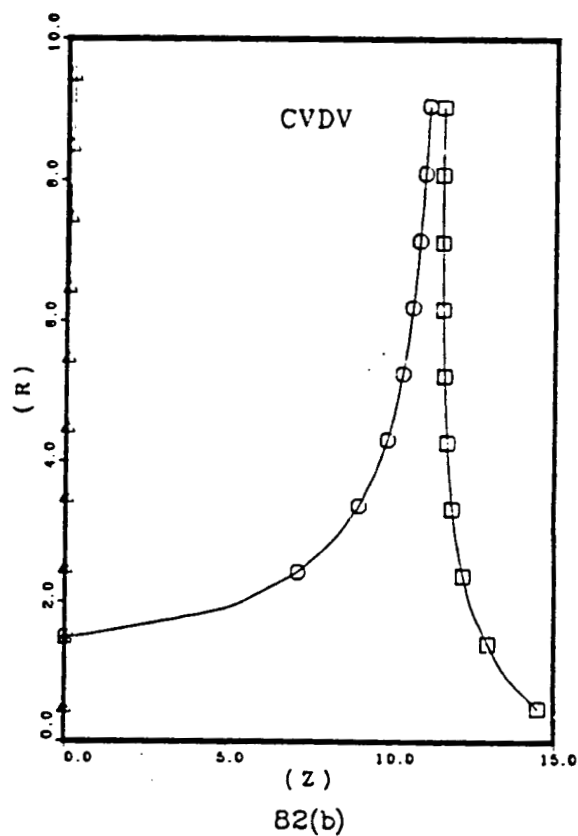
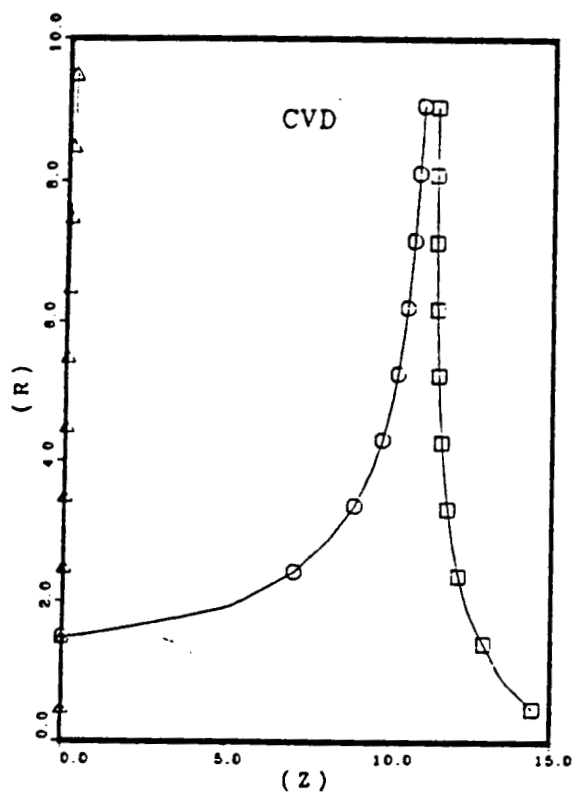
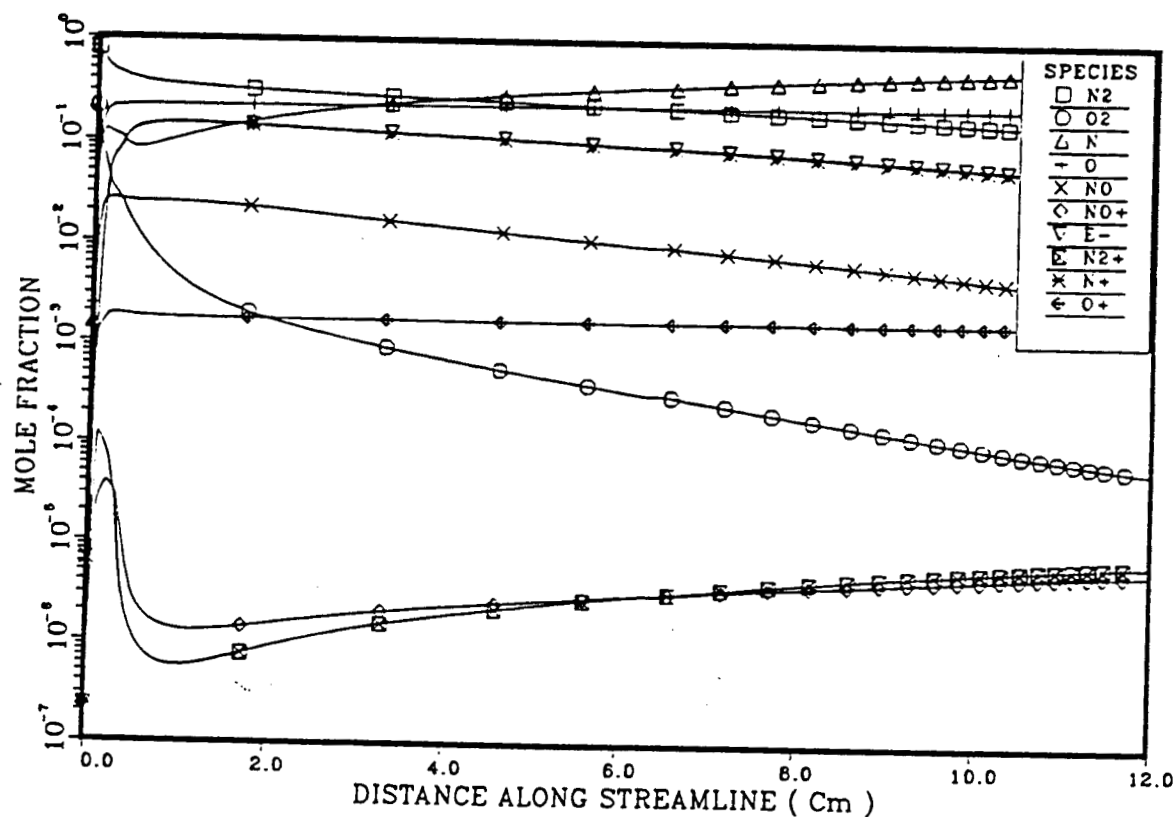


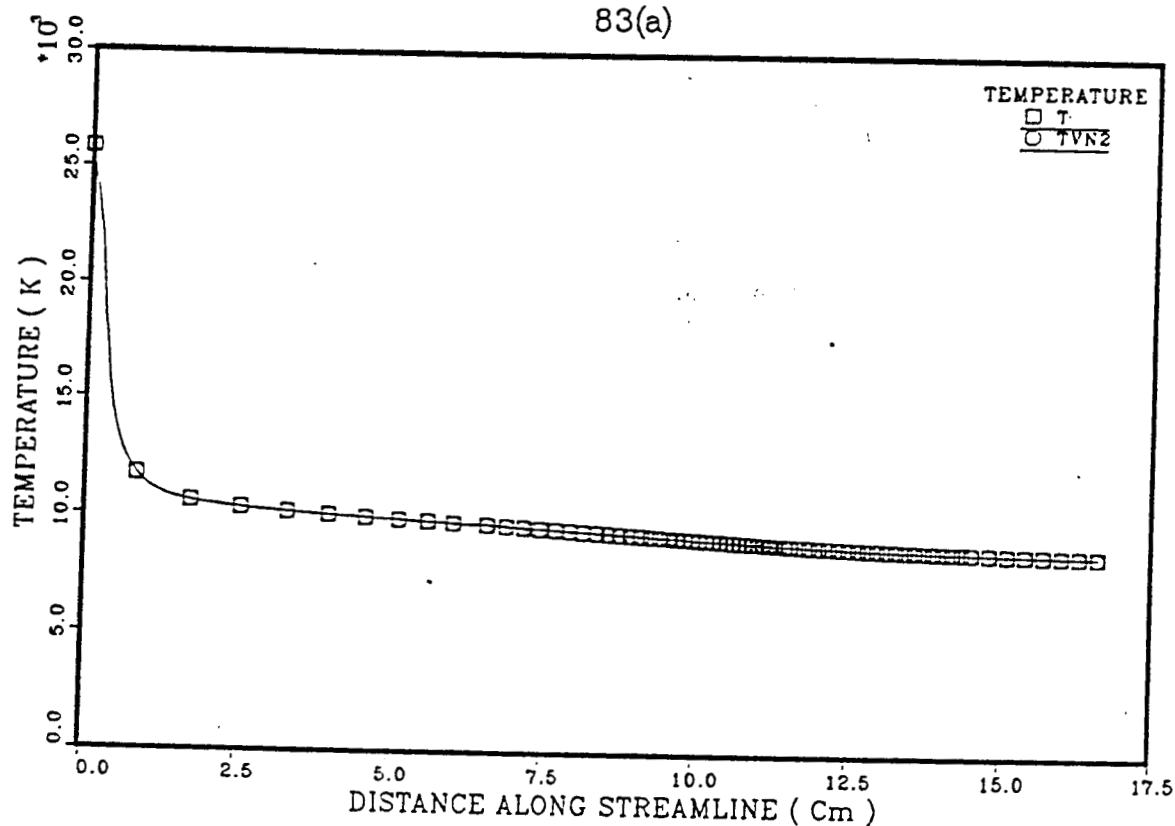
FIGURE 81.PROFILE AT V=10 Km/s, RR3



FIGURES 82(a),82(b),82(c),82(d).COORD,V=10 Km/s, RR3

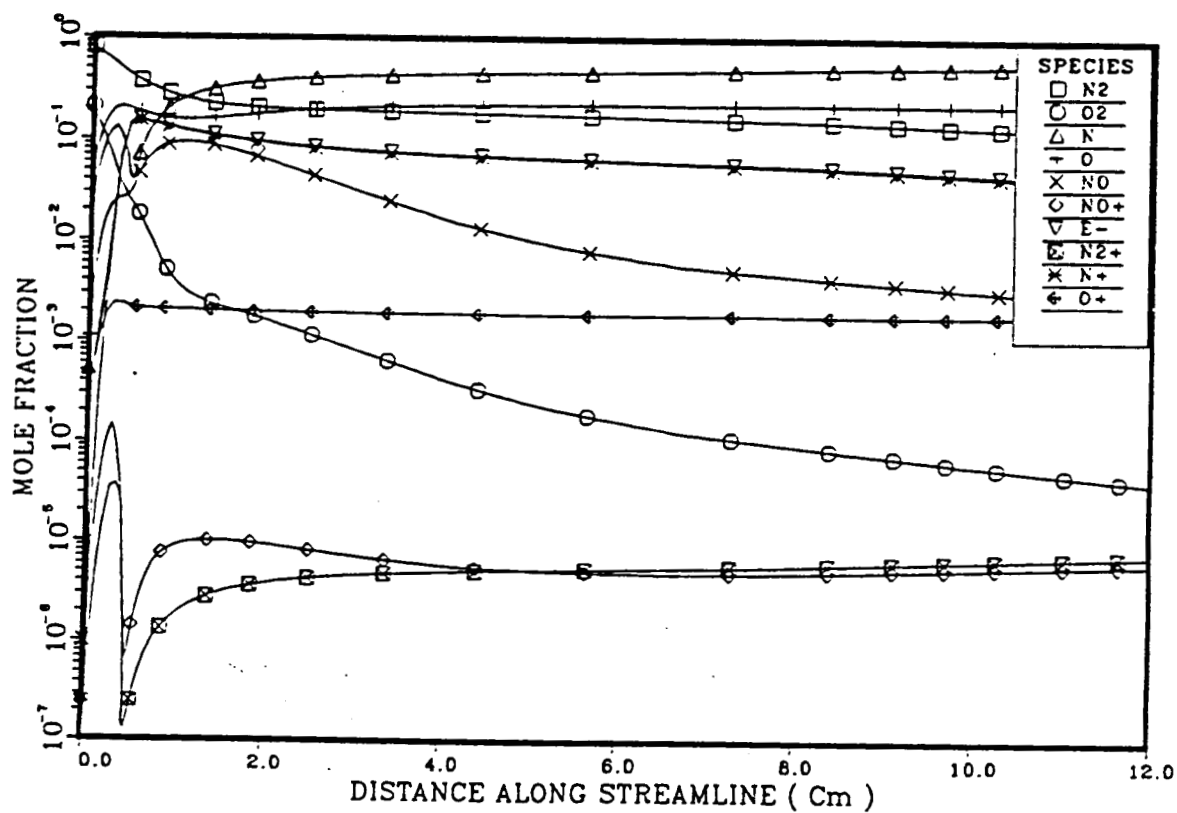


83(a)

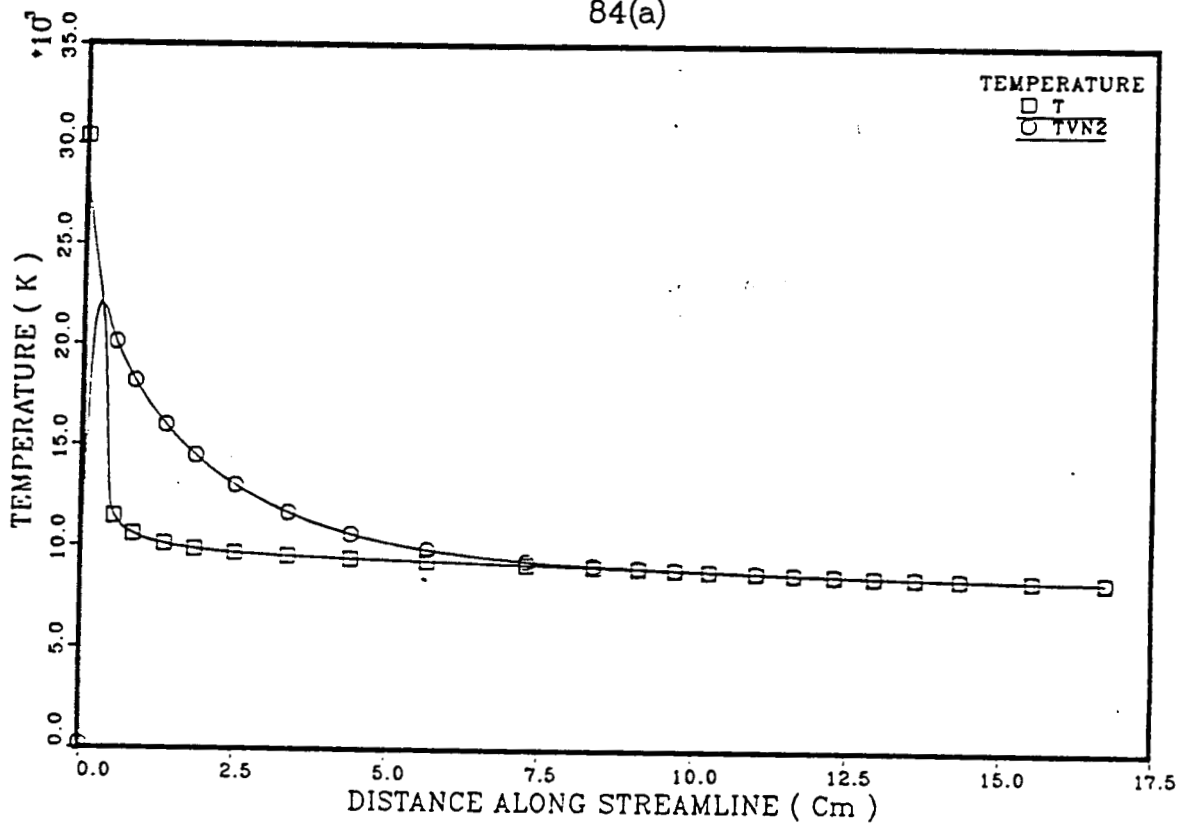


83(b)

FIGURES 83(a),83(b).VEQ MODEL AT V=8.9 Km/s, RR3

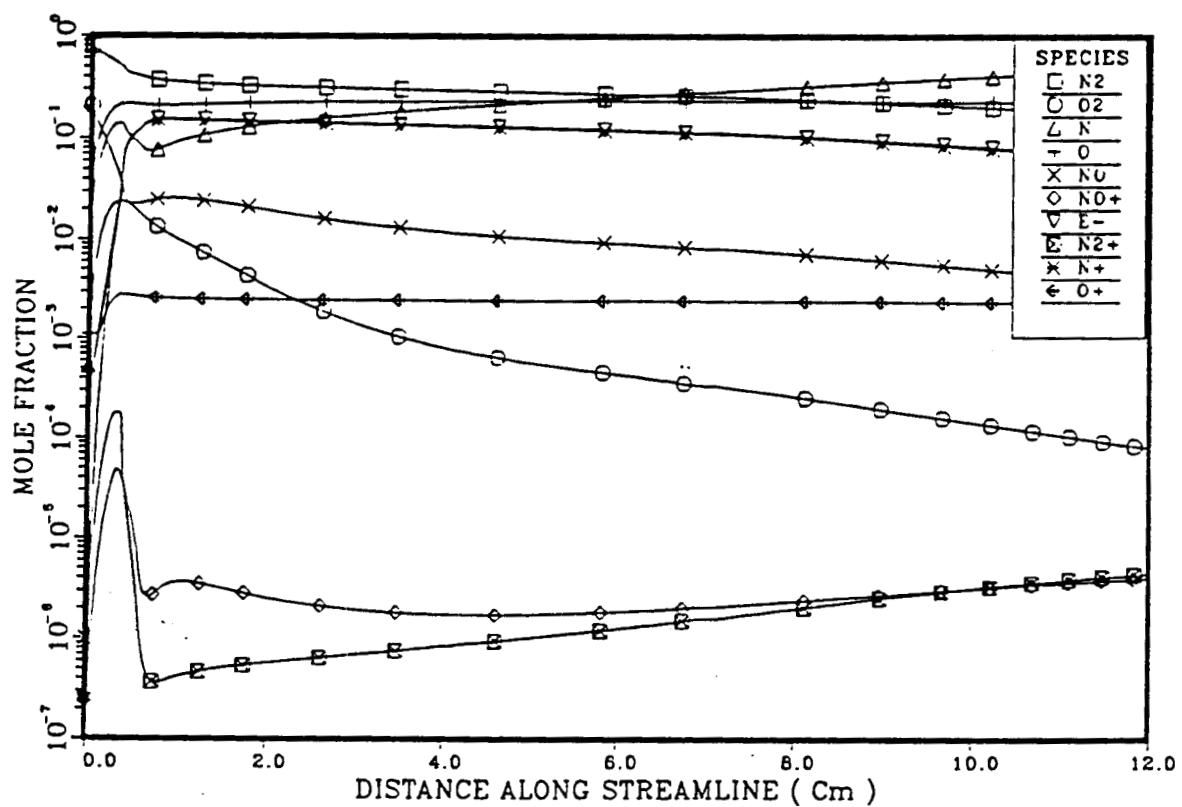


84(a)

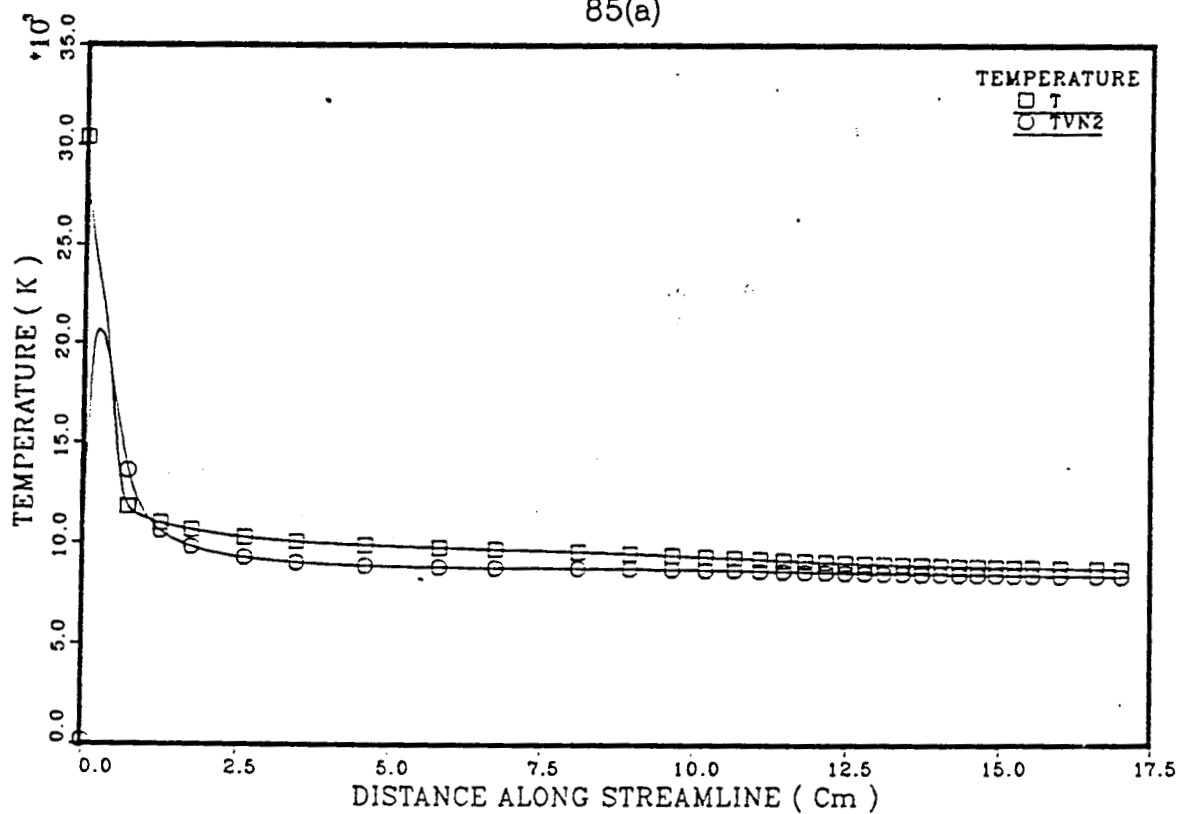


84(b)

FIGURES 84(a),84(b).CVD MODEL AT $V=8.9$ Km/s, RR3

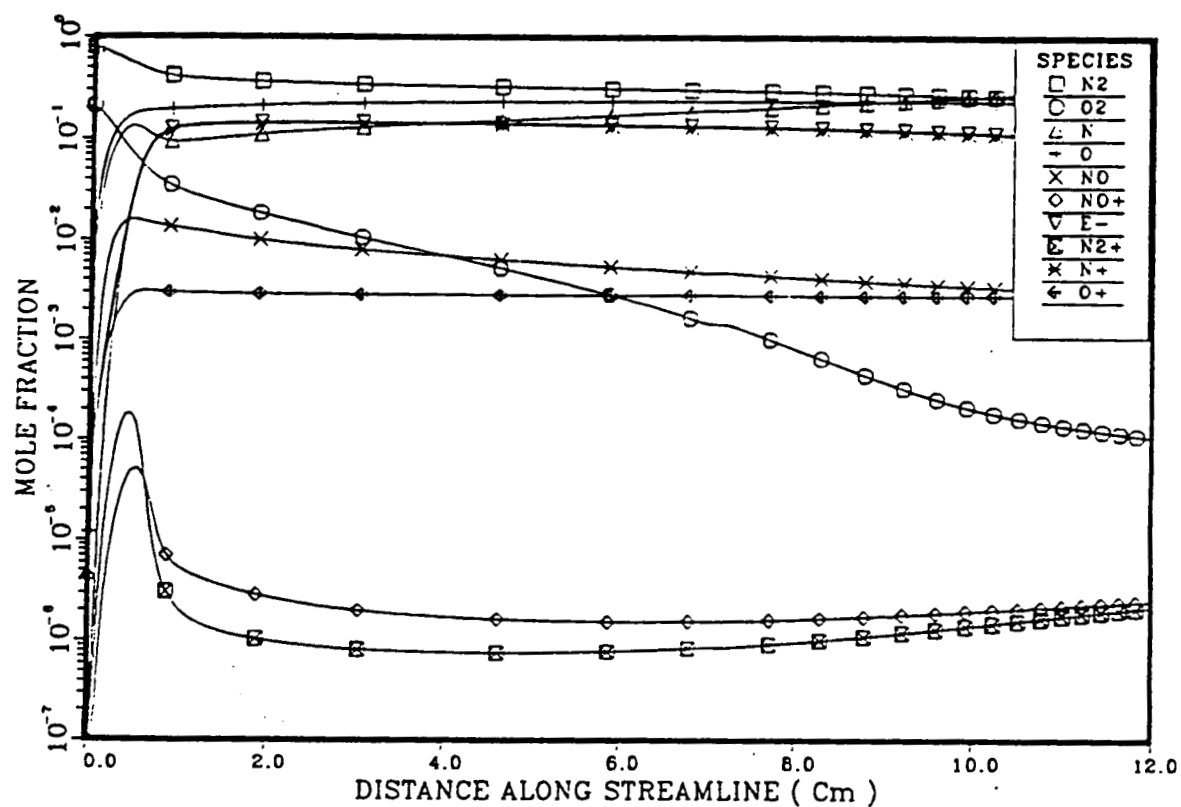


85(a)

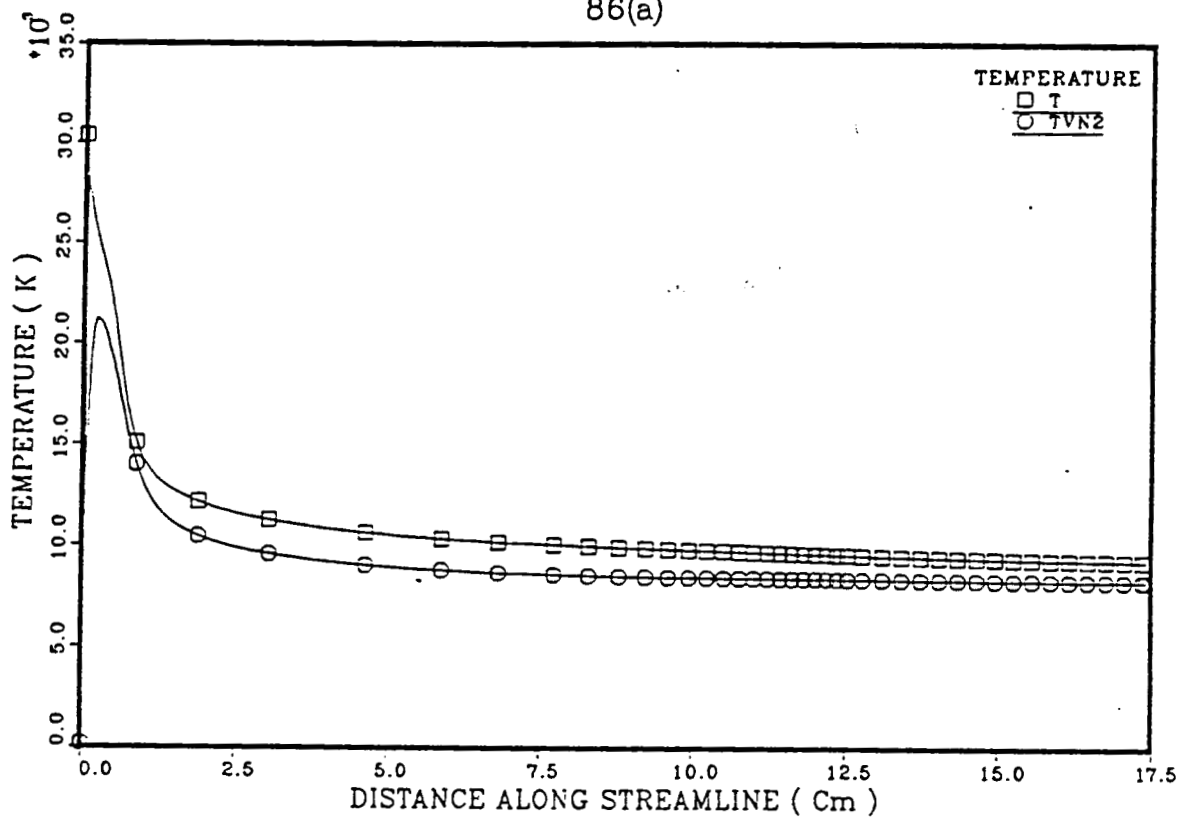


85(b)

FIGURES 85(a),85(b).CVDV MODEL AT V=8.9 Km/s, RR3

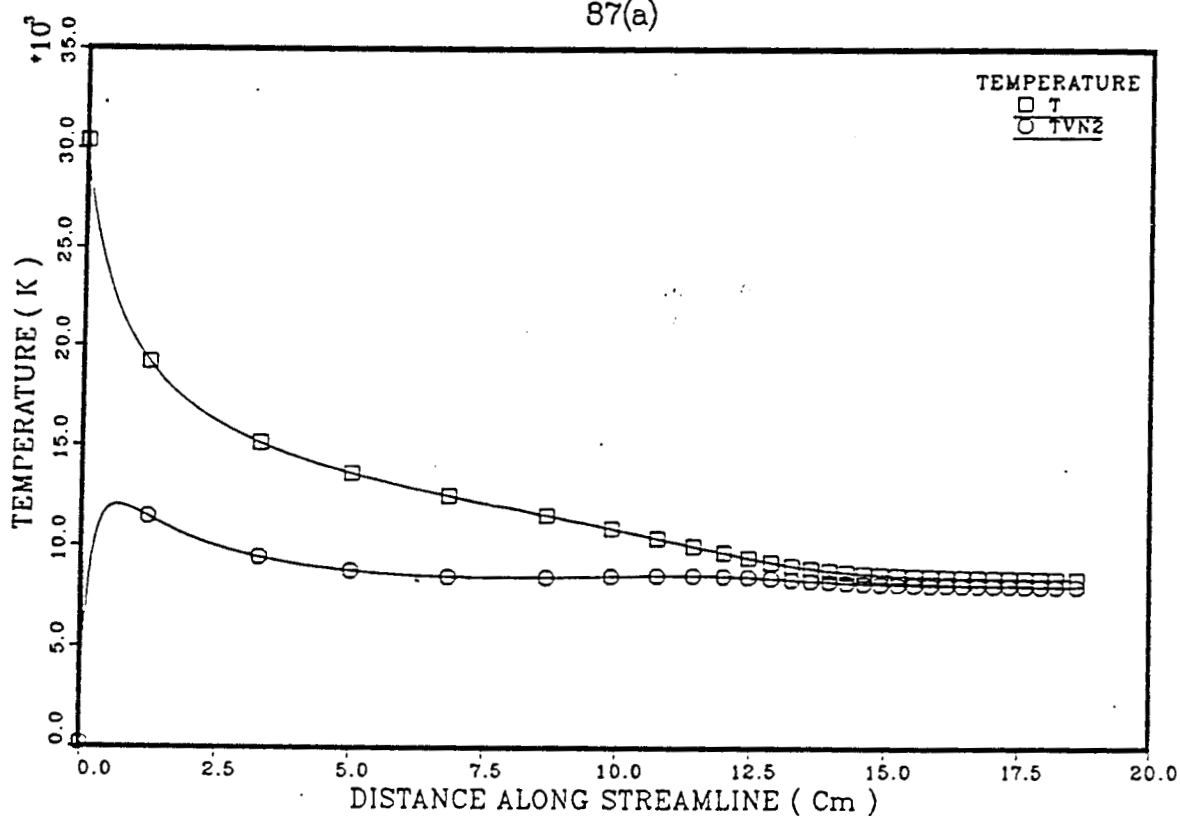
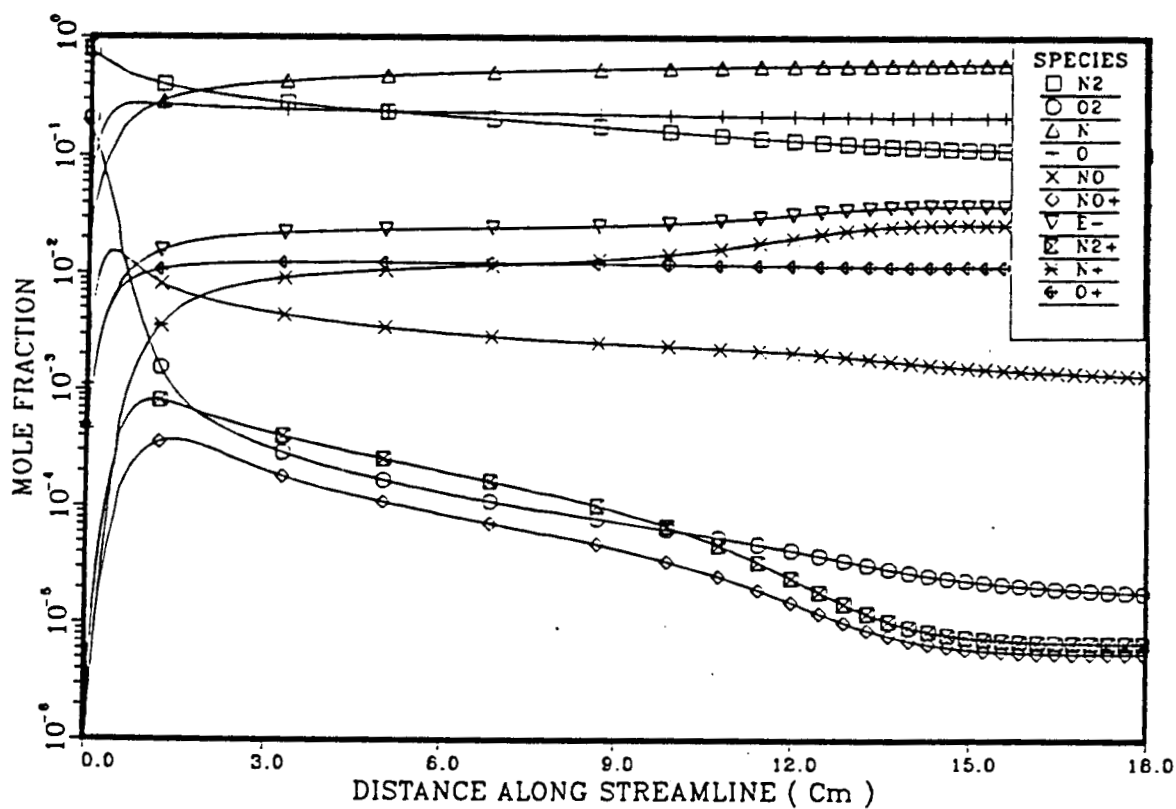


86(a)

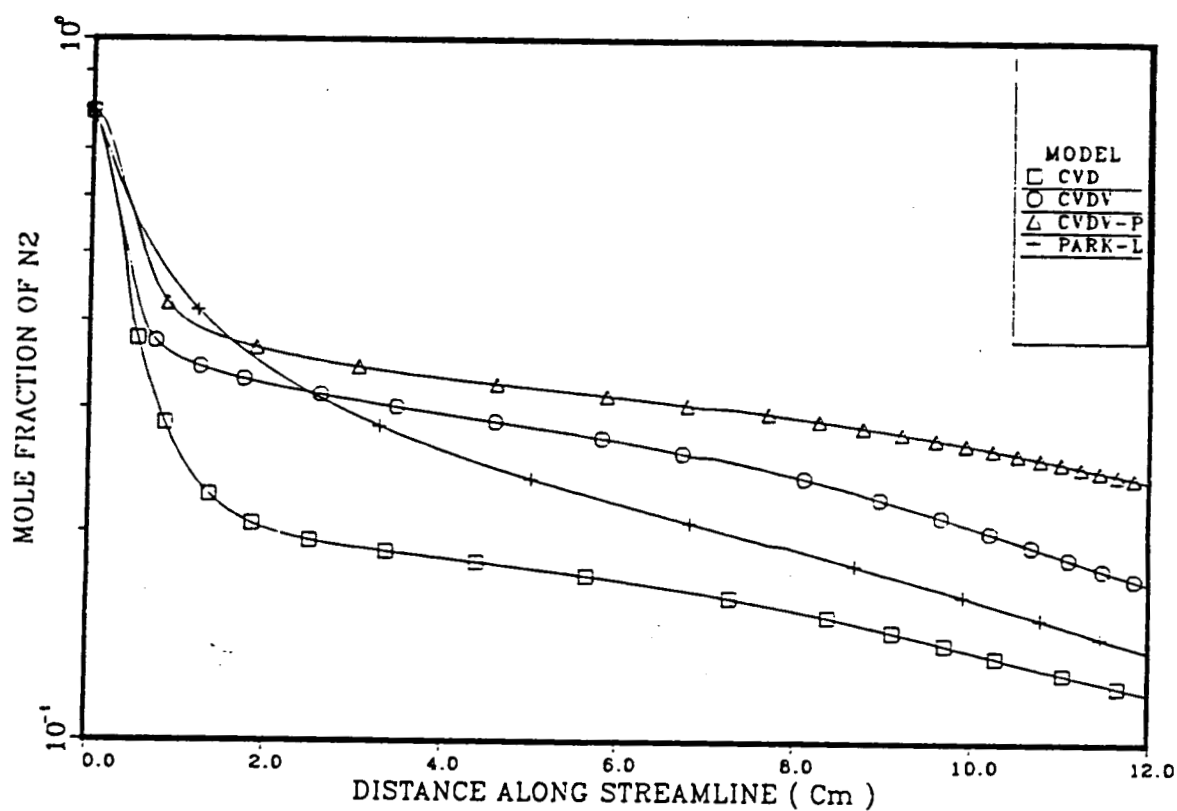


86(b)

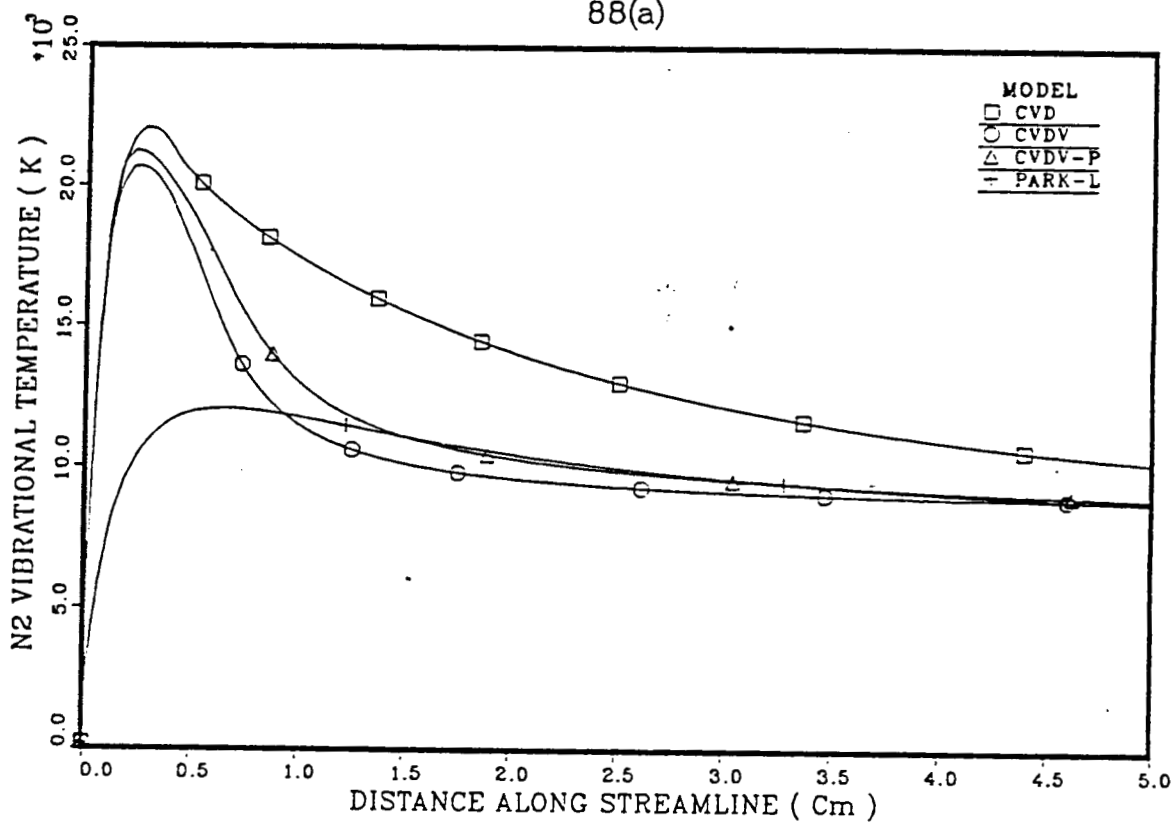
FIGURES 86(a),86(b).CVDV-P MODEL AT V=8.9 Km/s, RR3



FIGURES 87(a),87(b).PARK-L MODEL AT V=8.9 Km/s, RR3

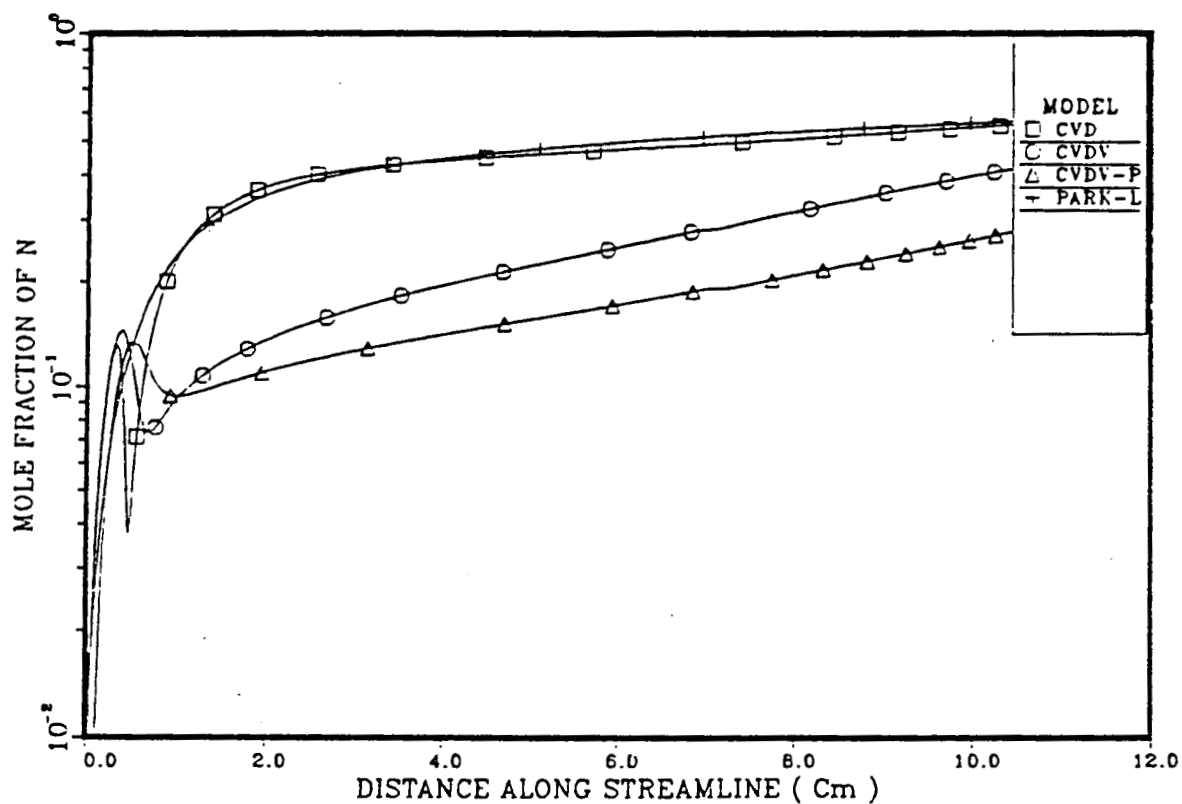


88(a)

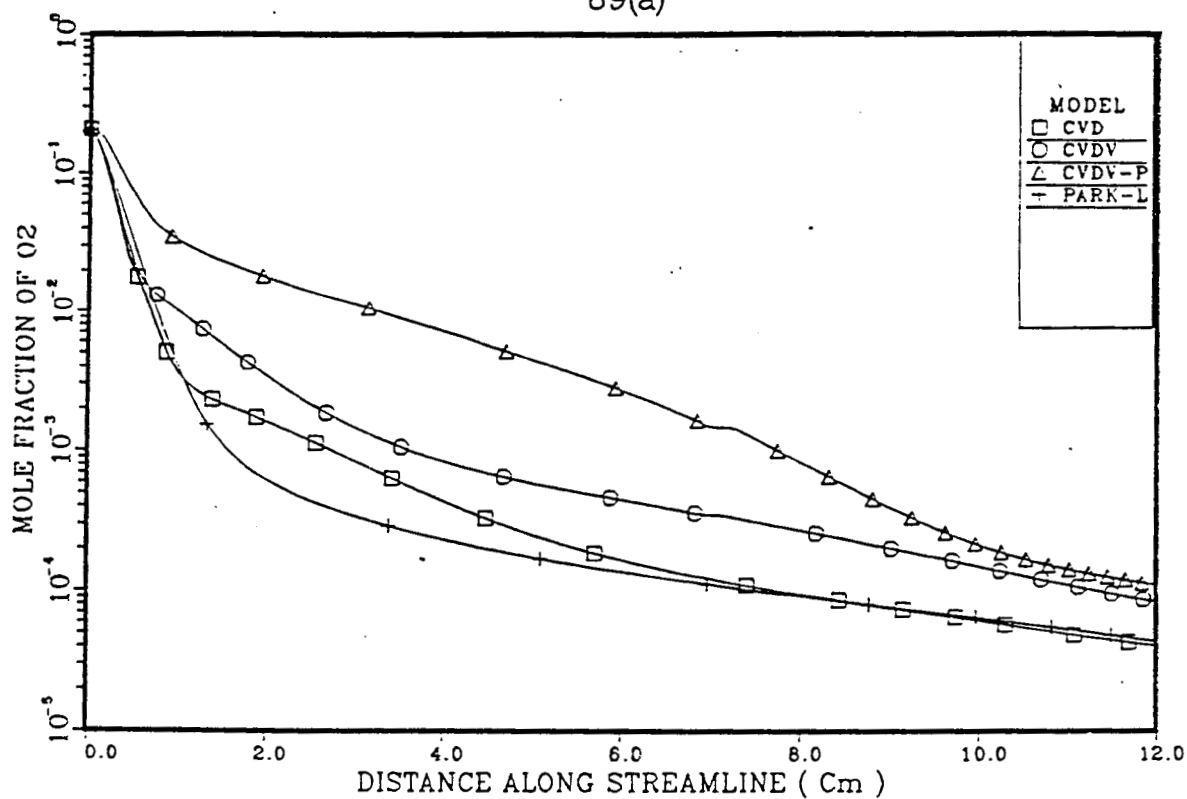


88(b)

FIGURES 88(a),88(b).PROFILES AT V=8.9 Km/s, RR3

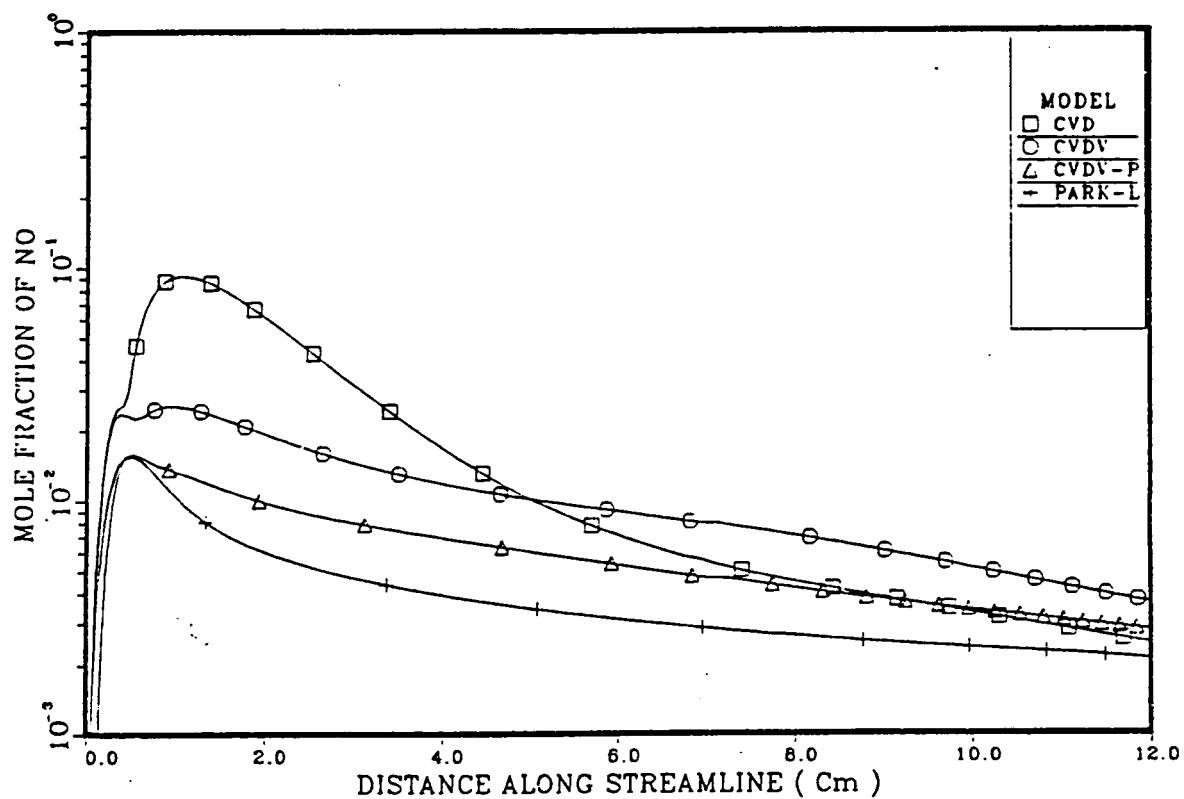


89(a)

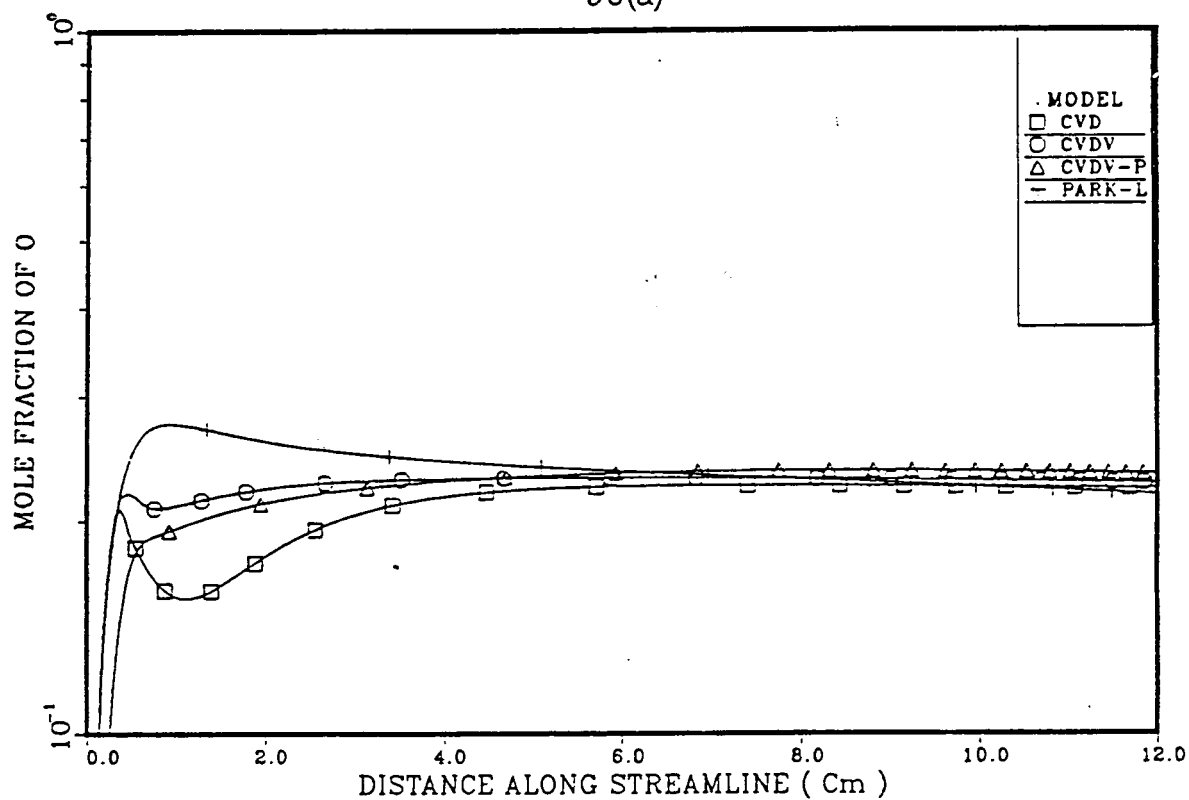


89(b)

FIGURES 89(a),89(b).PROFILES AT V=8.9 Km/s, RR3

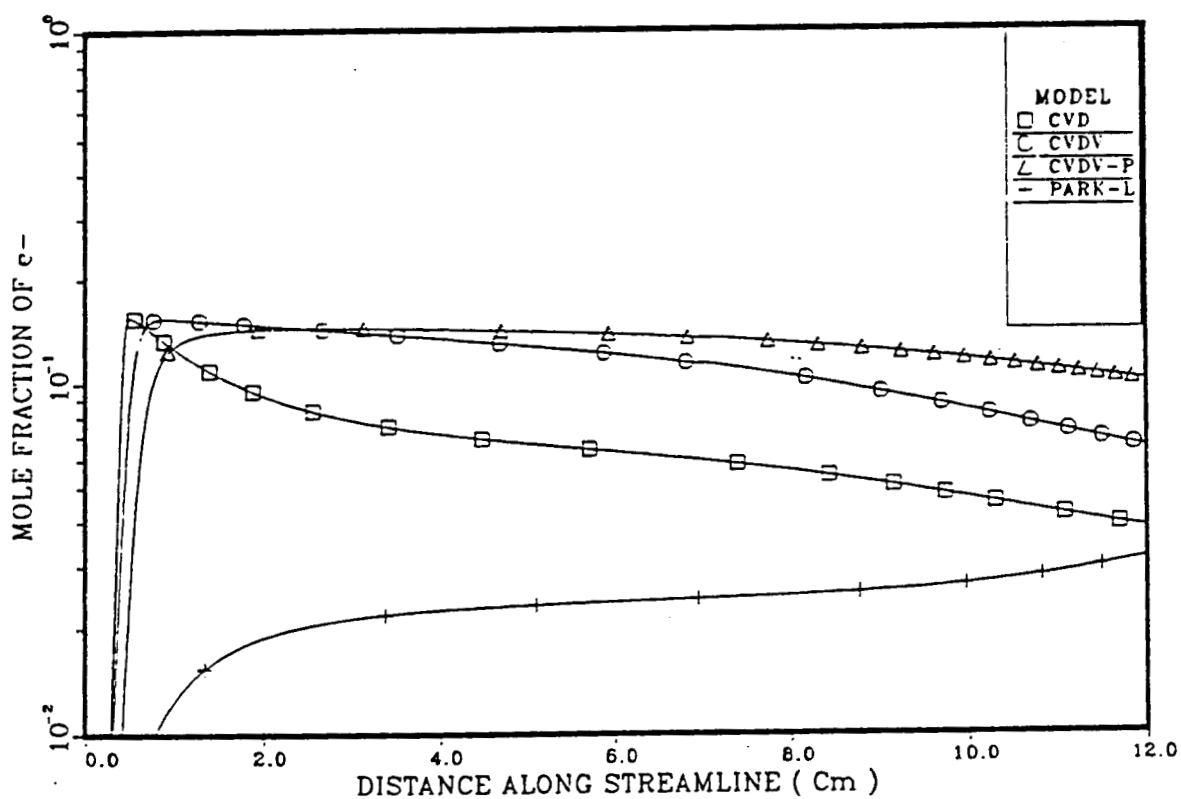


90(a)

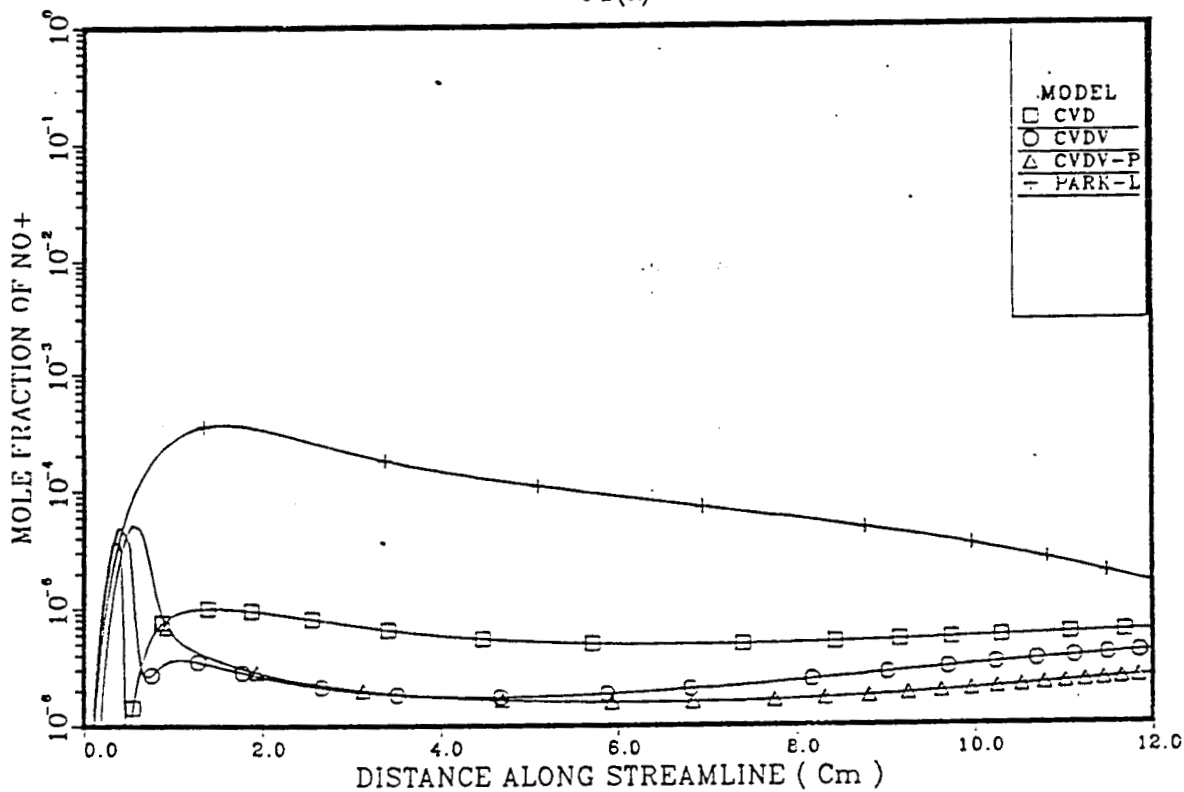


90(b)

FIGURES 90(a),90(b).PROFILES AT $V=8.9$ Km/s, RR3

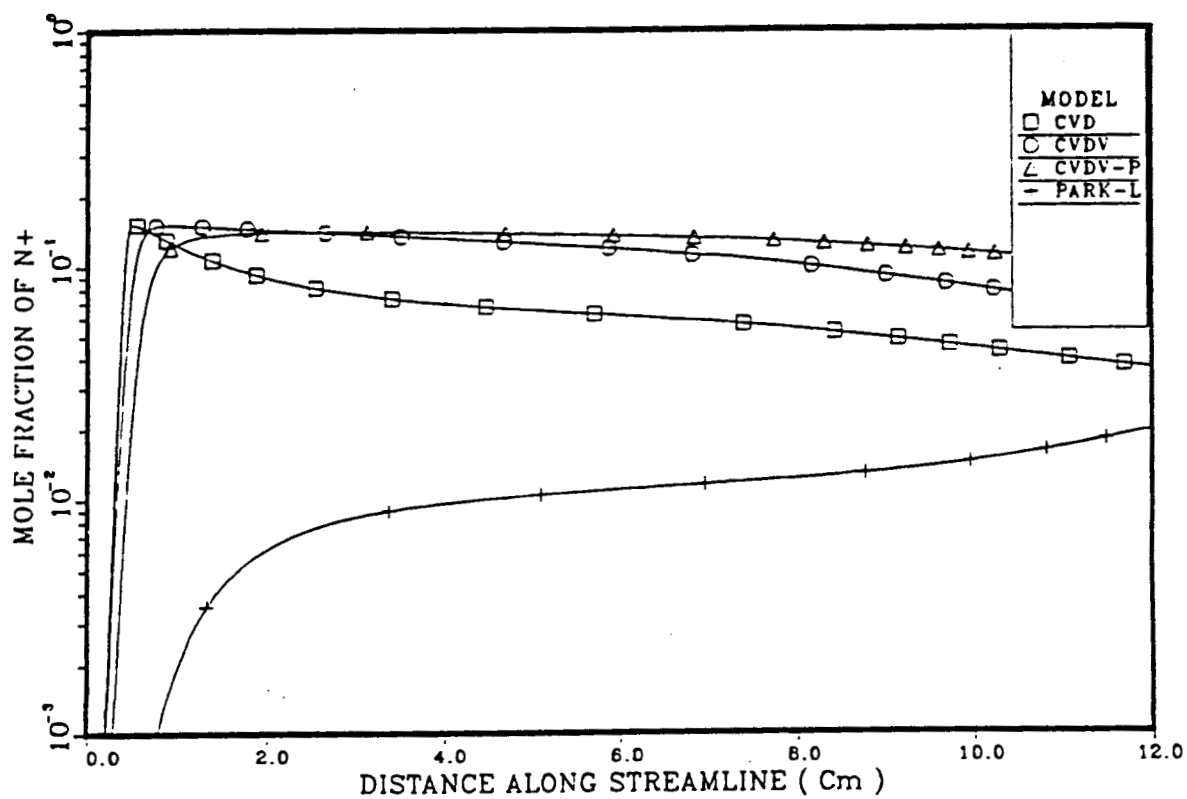


91(a)

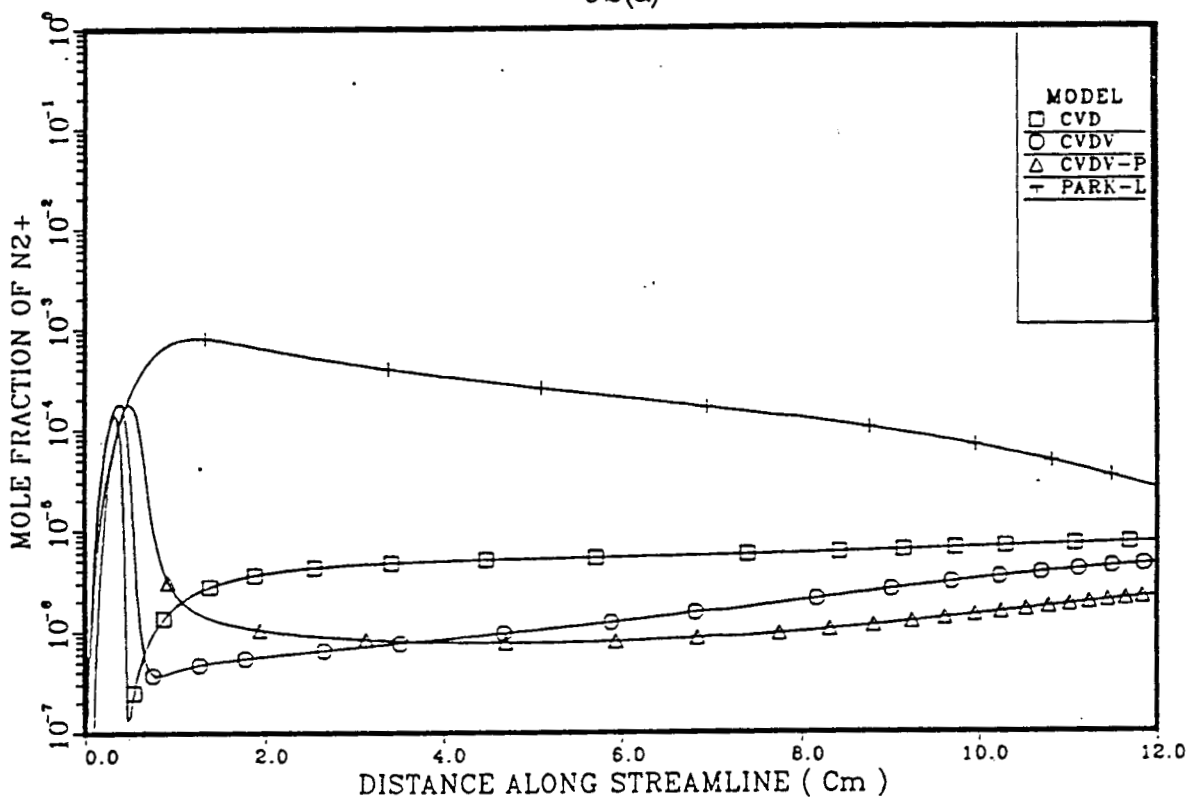


91(b)

FIGURES 91(a),91(b).PROFILES AT $V=8.9$ Km/s, RR3



92(a)



92(b)

FIGURES 92(a),92(b).PROFILES AT $V=8.9$ Km/s, RR3

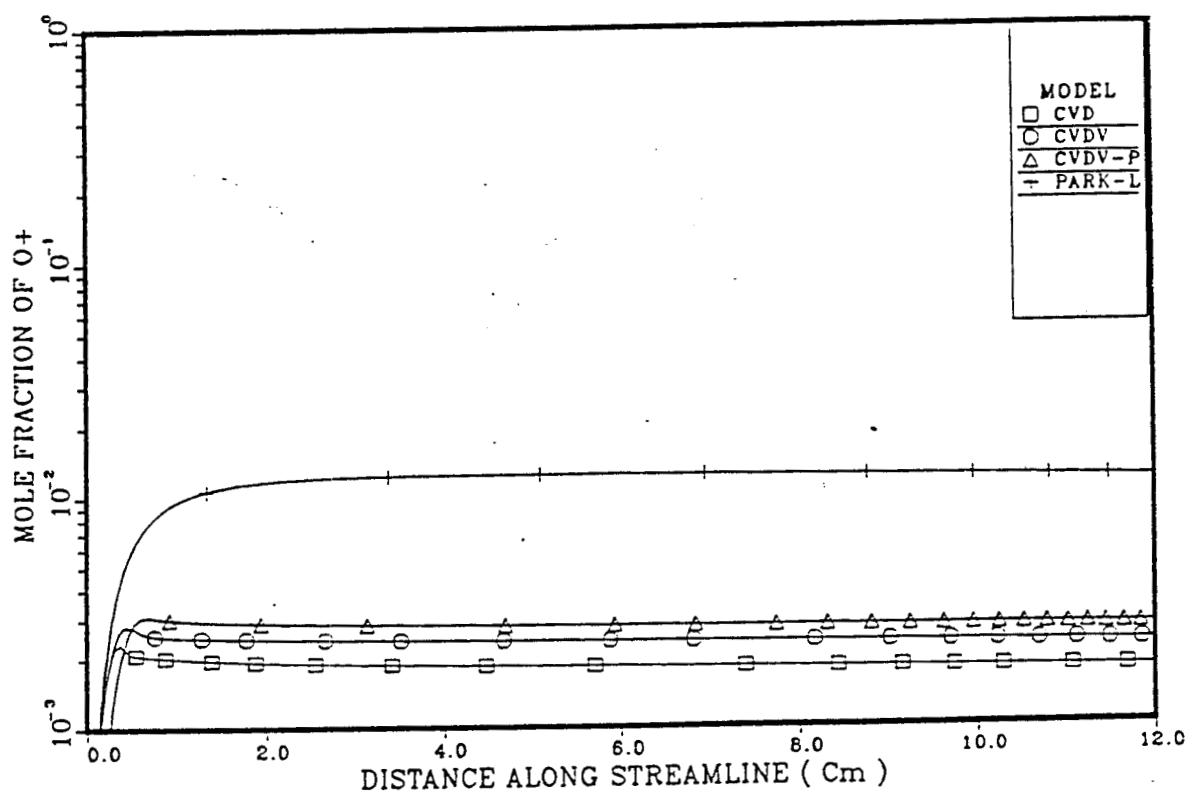
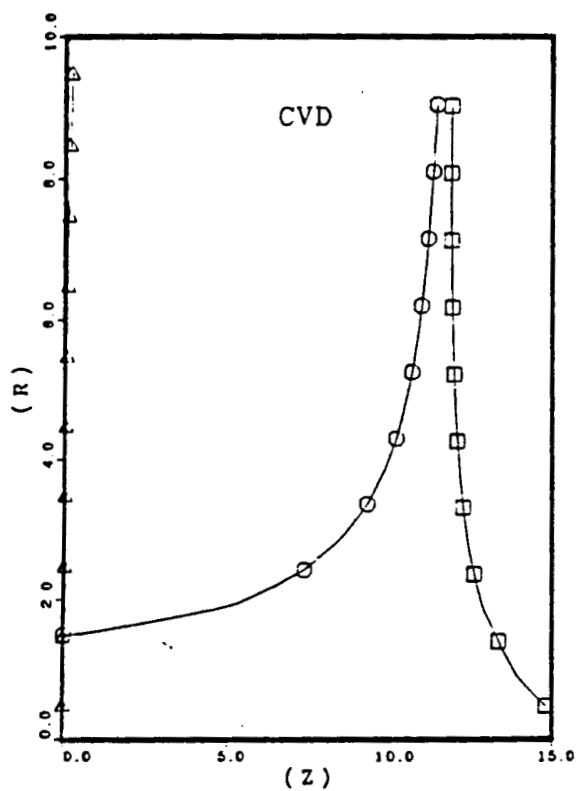
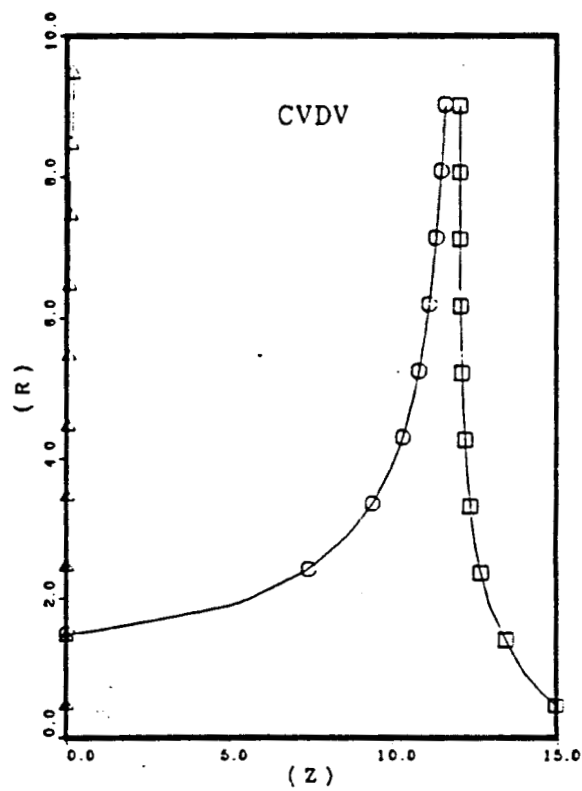


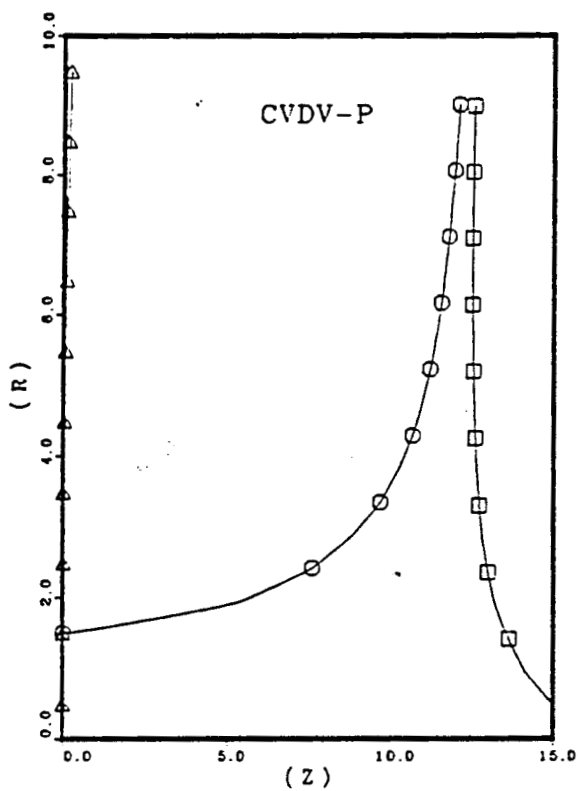
FIGURE 93.PROFILE AT V=8.9 Km/s, RR3



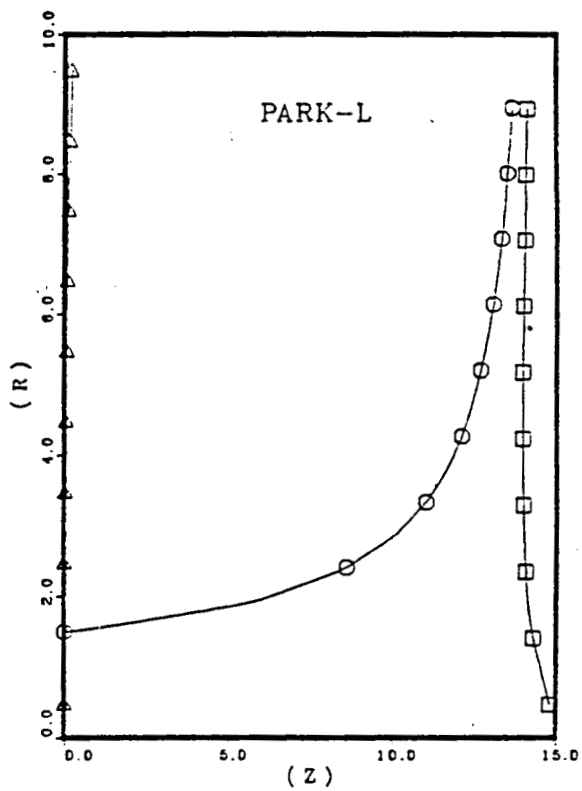
94(a)



94(b)



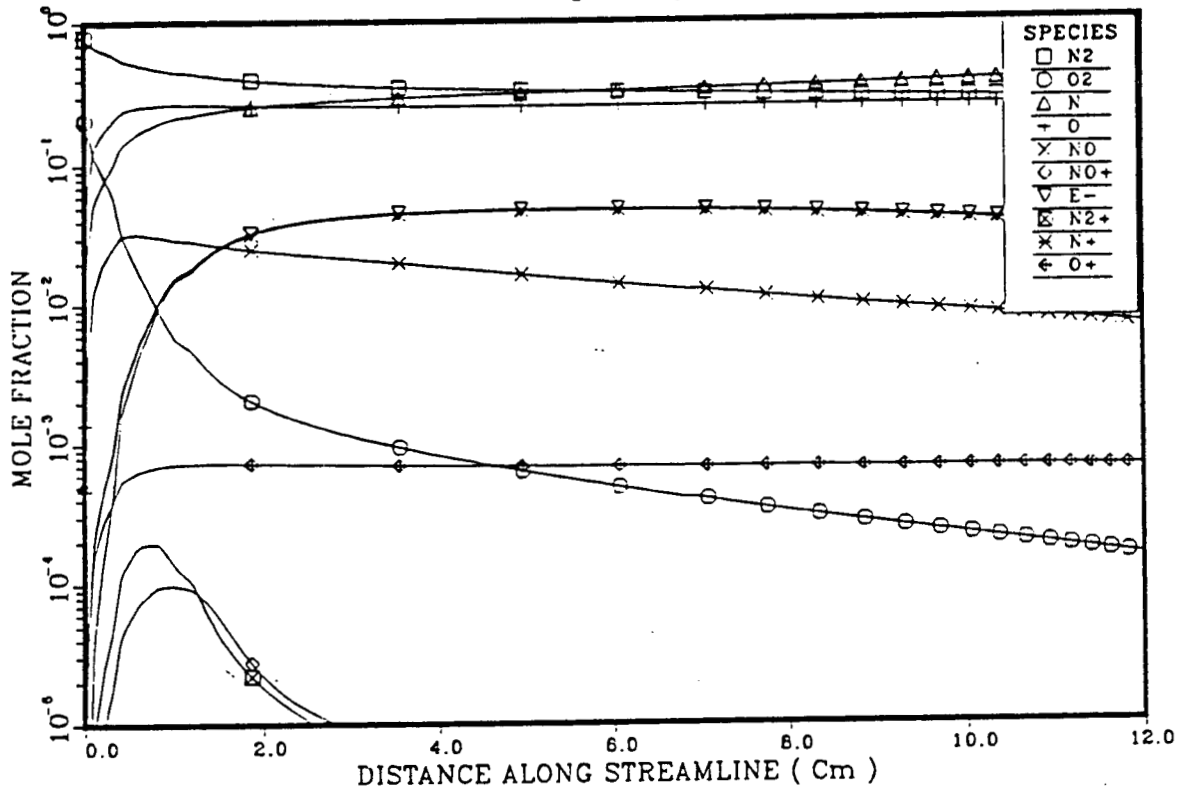
94(c)



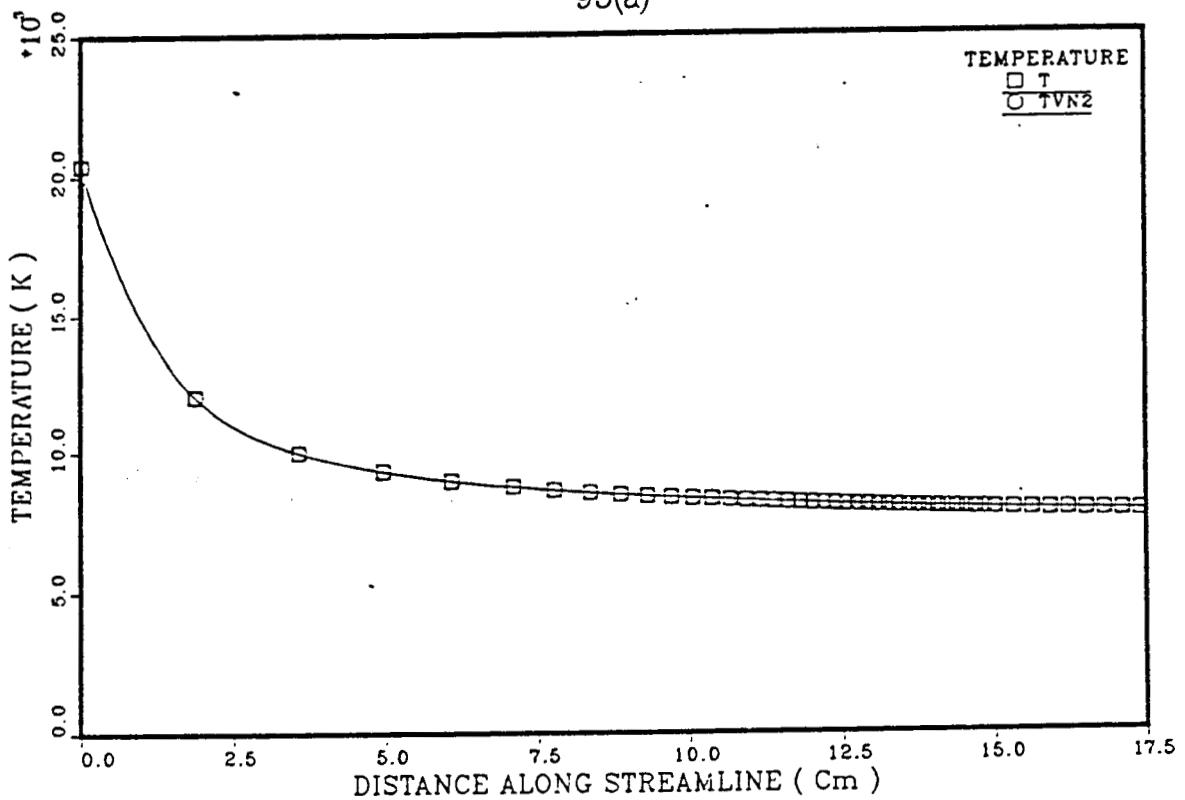
94(d)

FIGURES 94(a),94(b),94(c),94(d).COORD,V=8.9 Km/s, RR3

ORIGINAL PAGE IS
OF POOR QUALITY

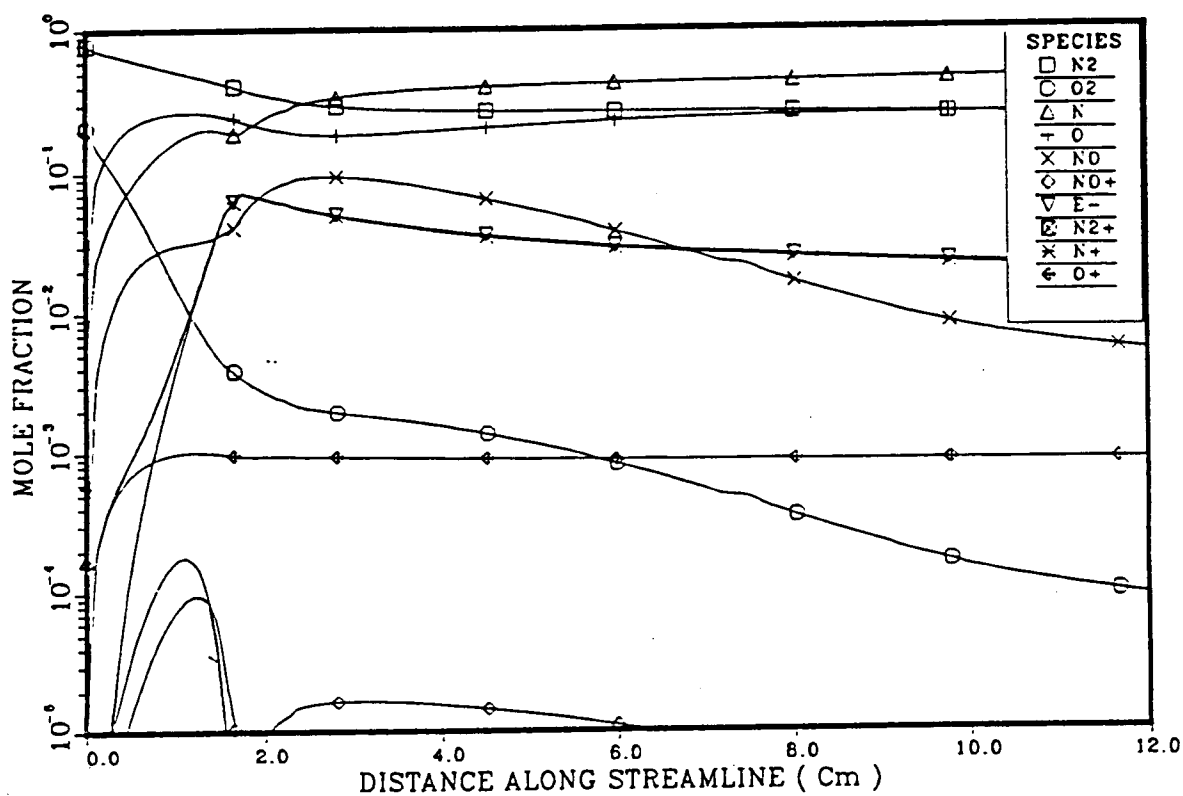


95(a)

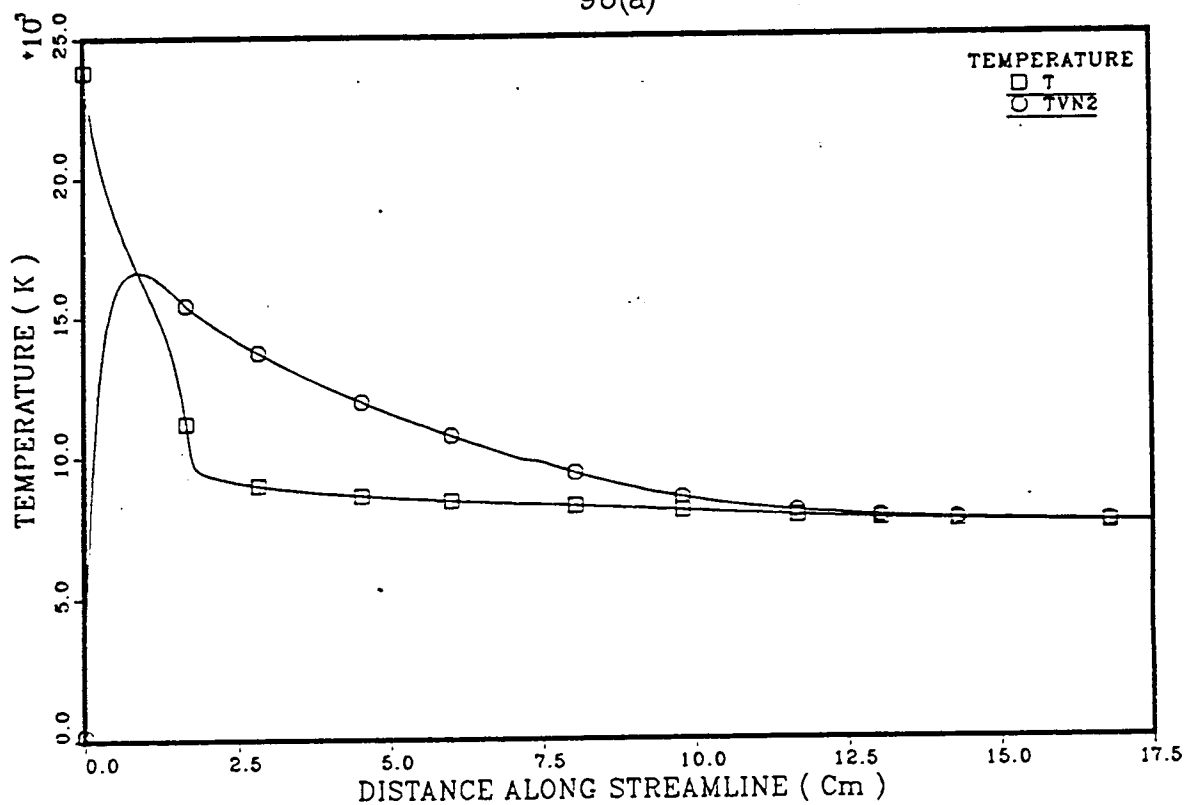


95(b)

FIGURES 95(a),95(b).VEQ MODEL AT V=7.7 Km/s, RR3



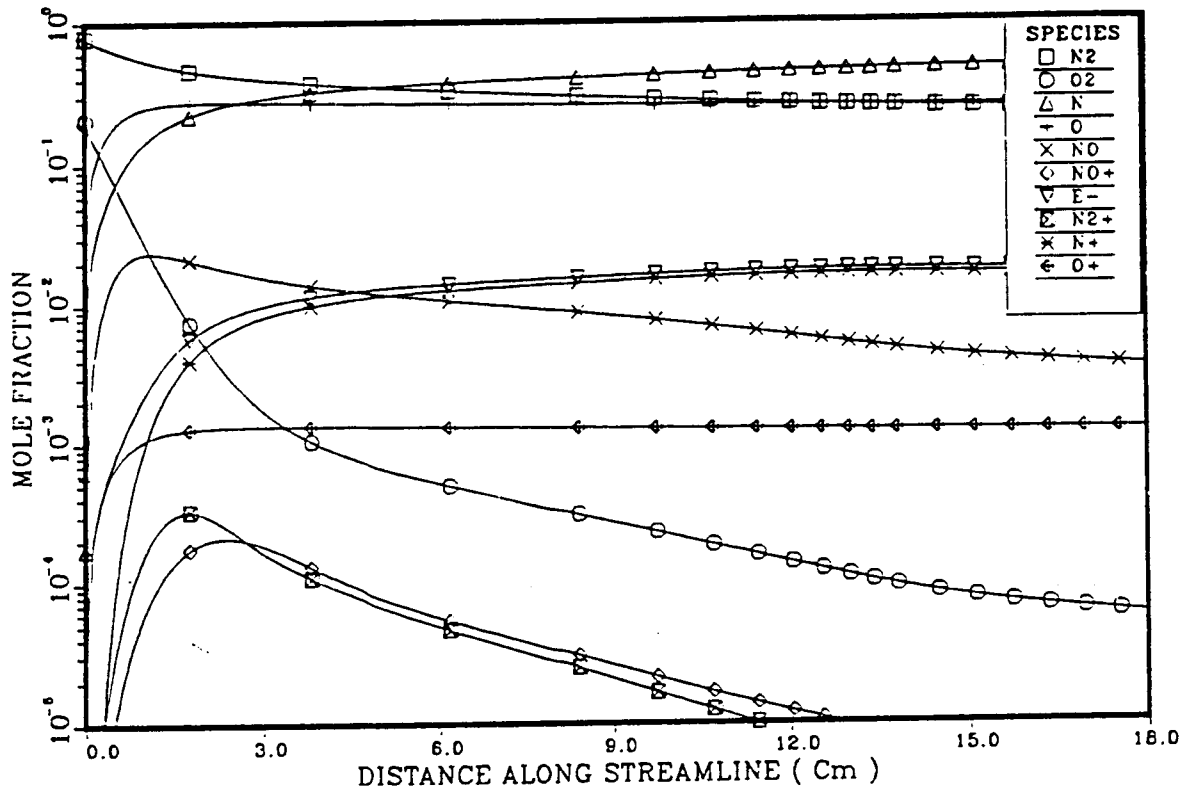
96(a)



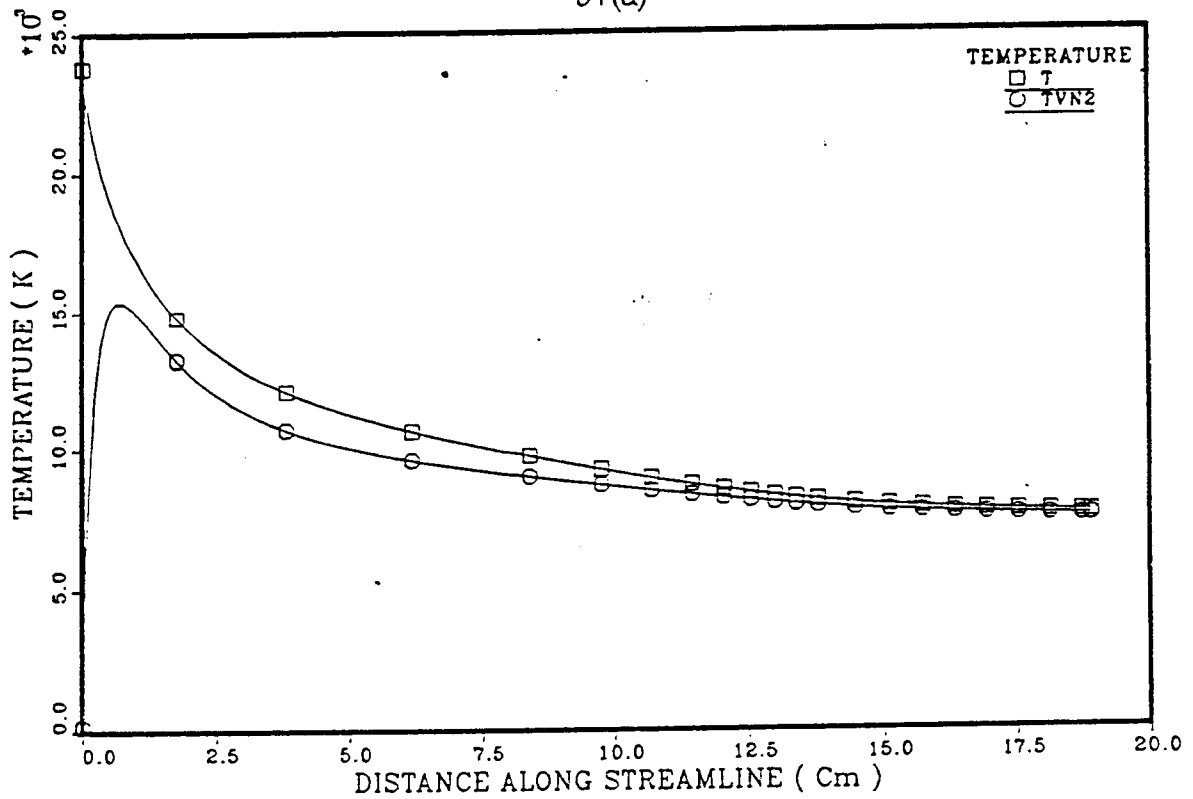
96(b)

FIGURES 96(a),96(b).CVD MODEL AT V=7.7 Km/s, RR3

ORIGINAL PAGE IS
OF POOR QUALITY

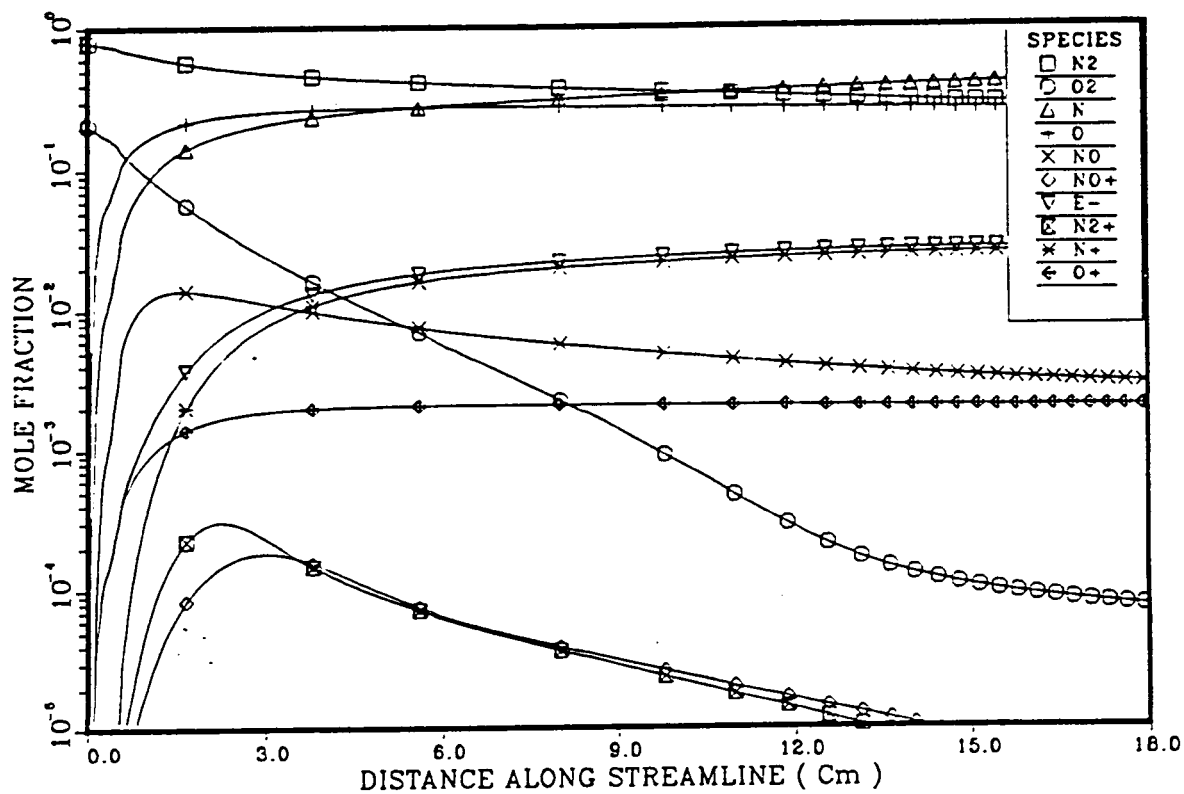


97(a)

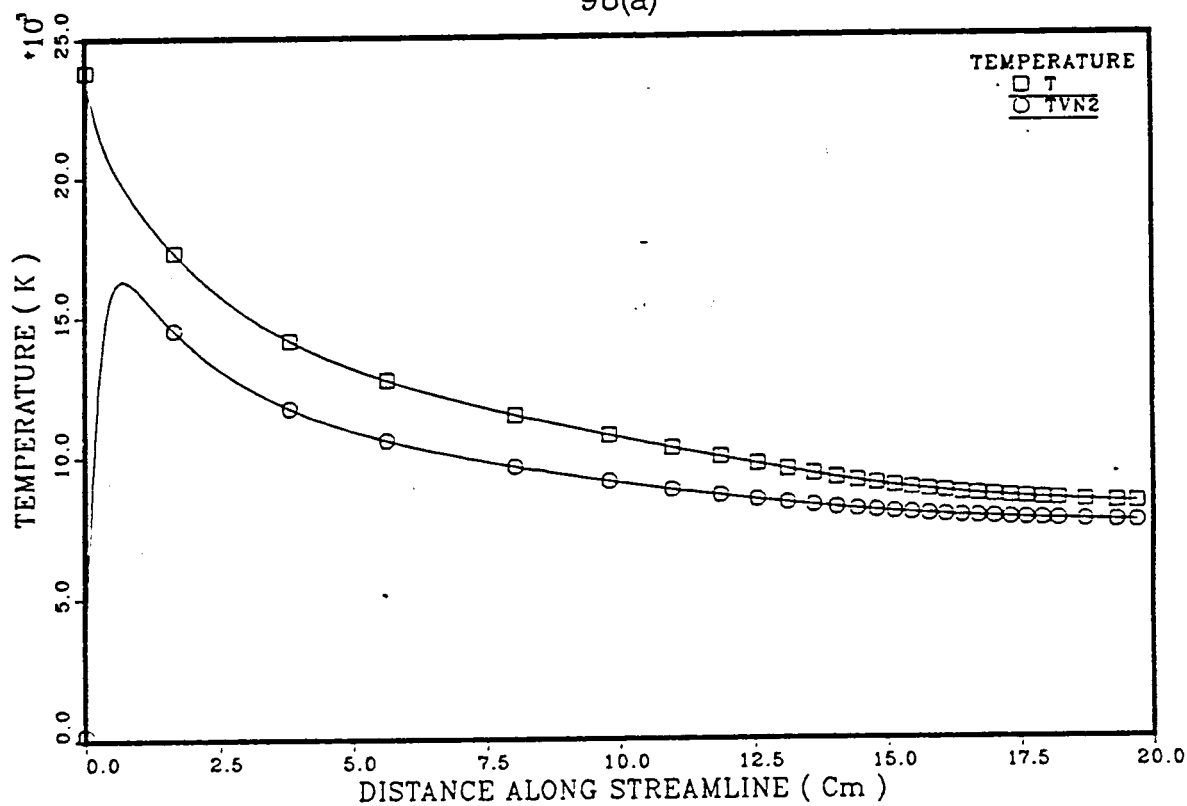


97(b)

FIGURES 97(a),97(b).CVDV MODEL AT $V=7.7$ Km/s, RR3

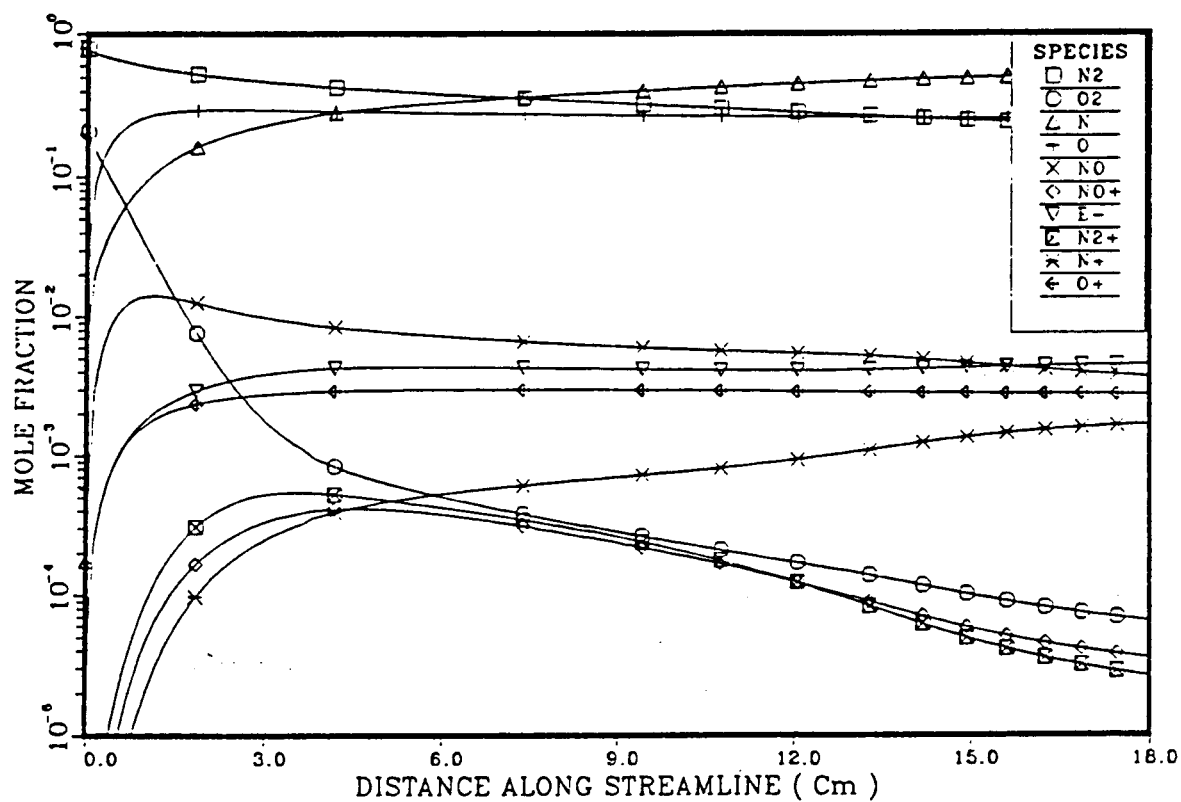


98(a)

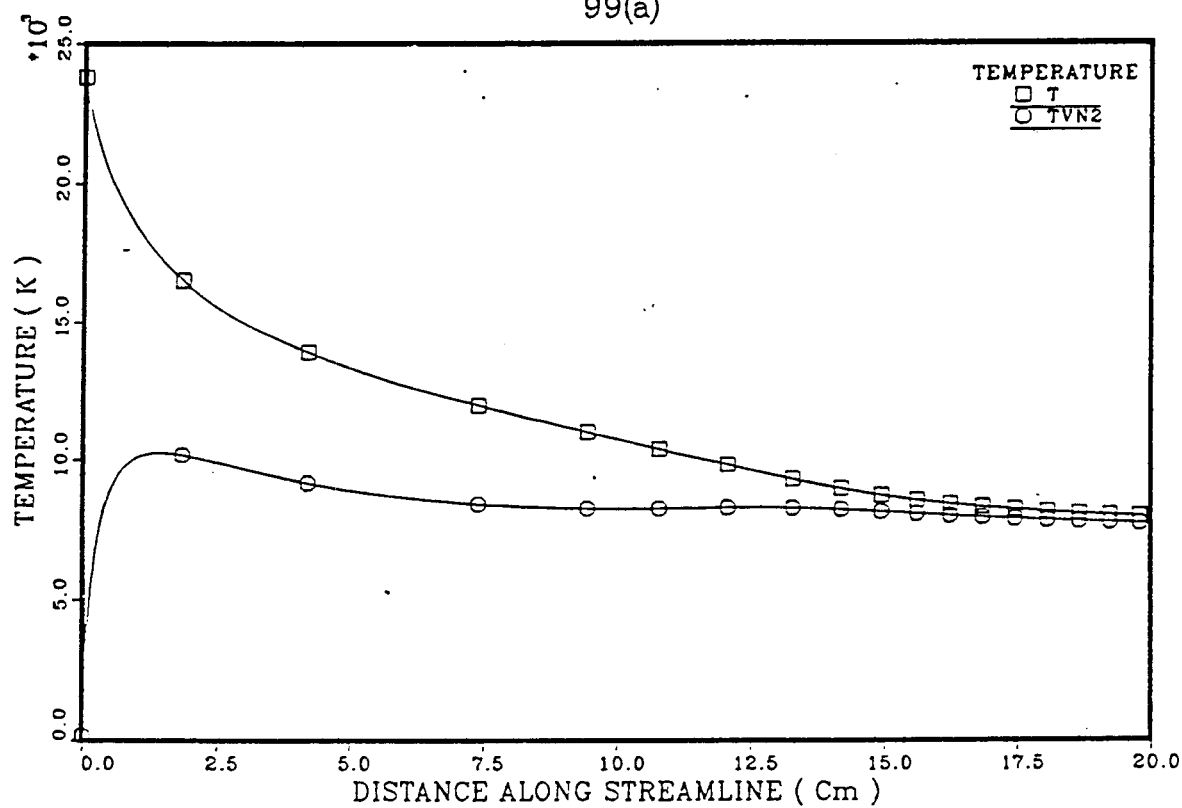


98(b)

FIGURES 98(a),98(b).CVDV-P MODEL AT V=7.7 Km/s, RR3

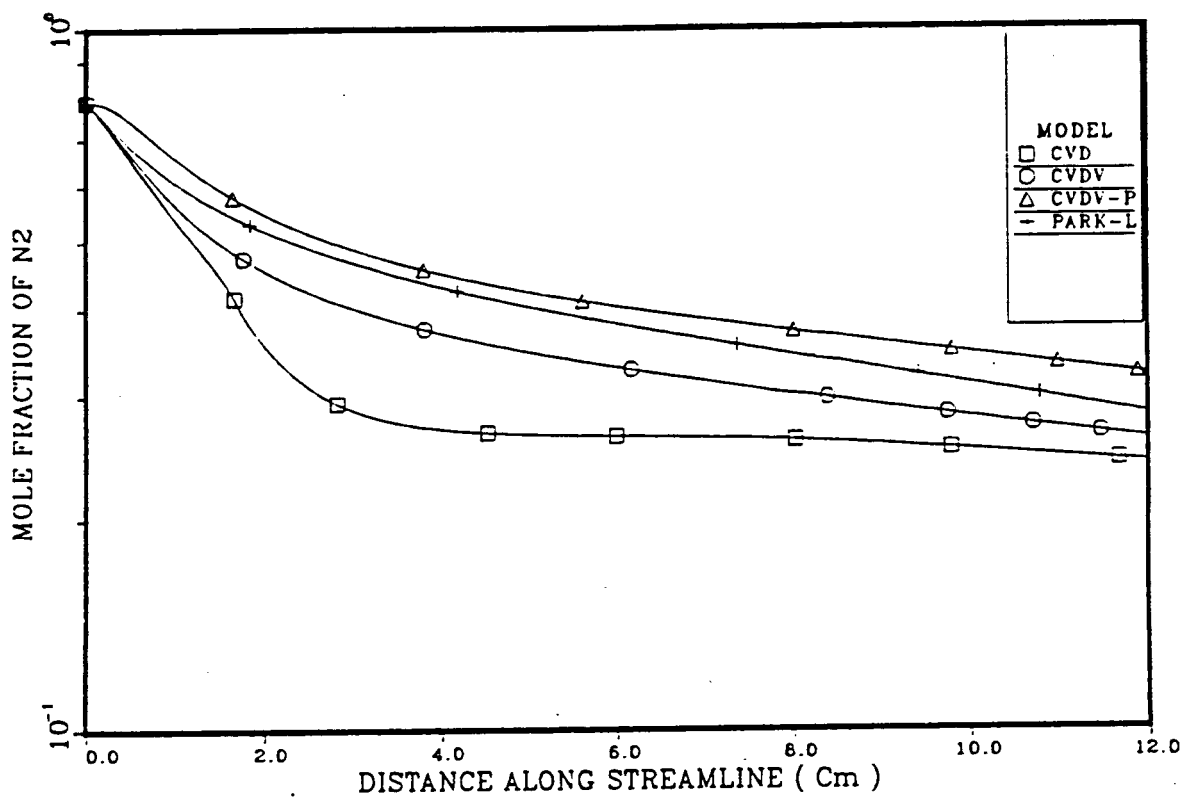


99(a)

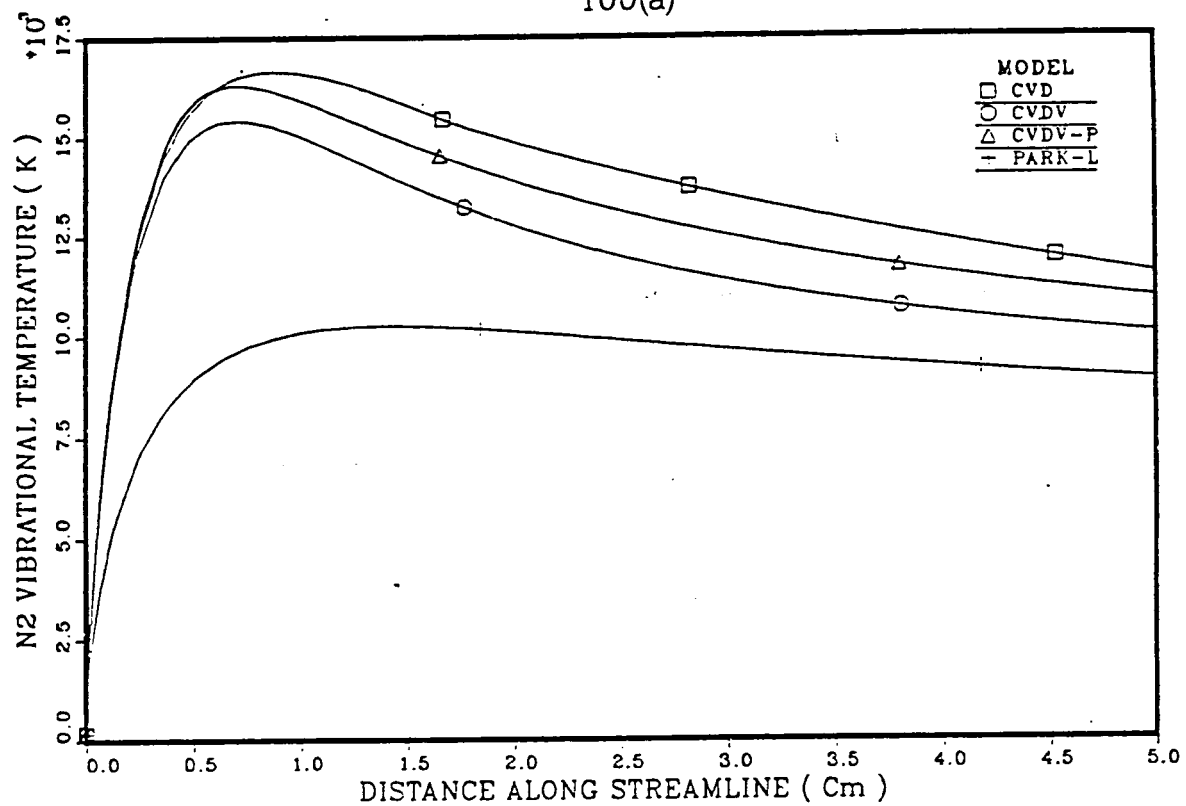


99(b)

FIGURES 99(a),99(b).PARK-L MODEL AT V=7.7 Km/s, RR3



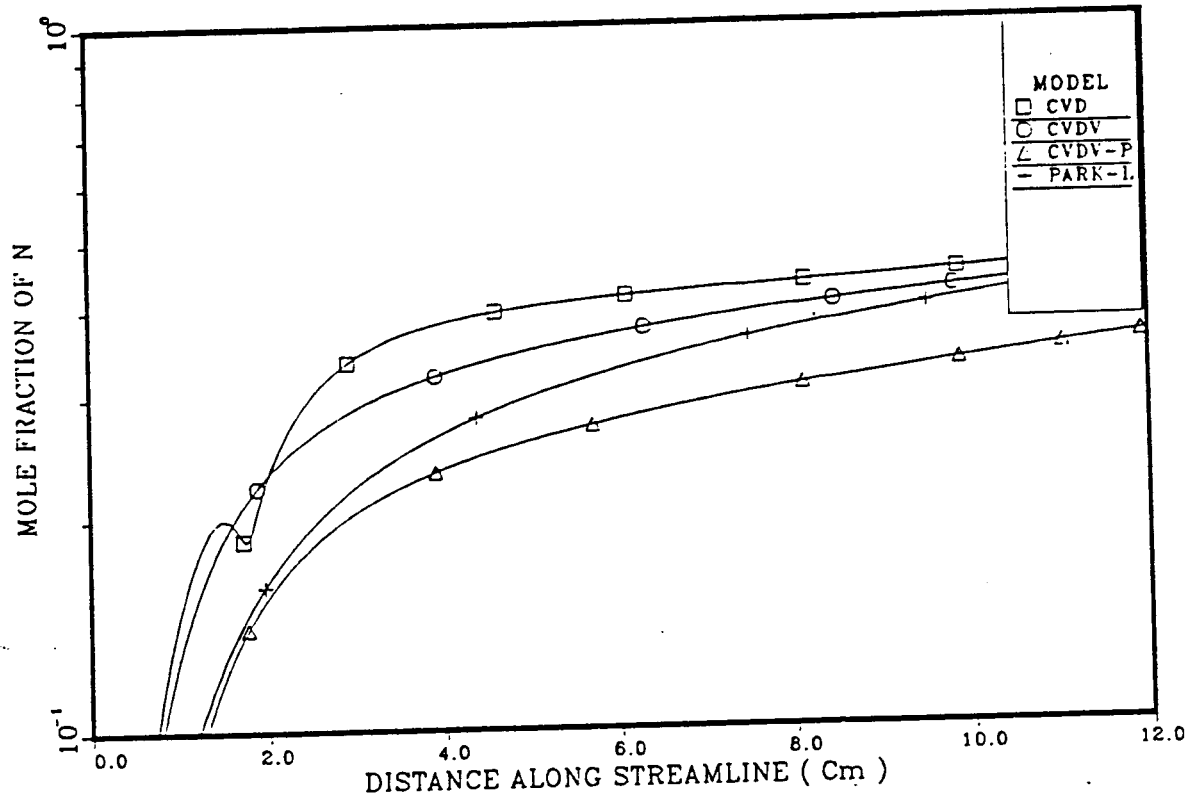
100(a)



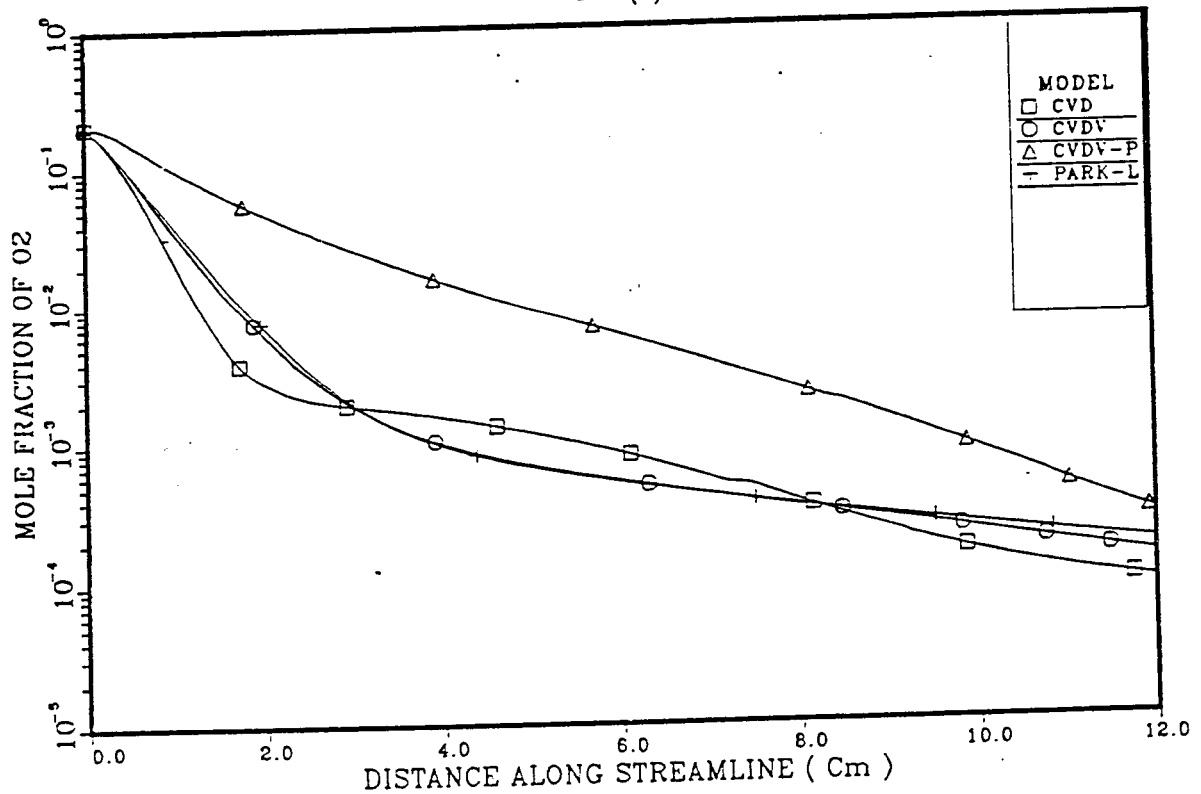
100(b)

FIGURES 100(a),100(b).PROFILES AT V=7.7 Km/s, RR3

ORIGINAL PAGE IS
OF POOR QUALITY

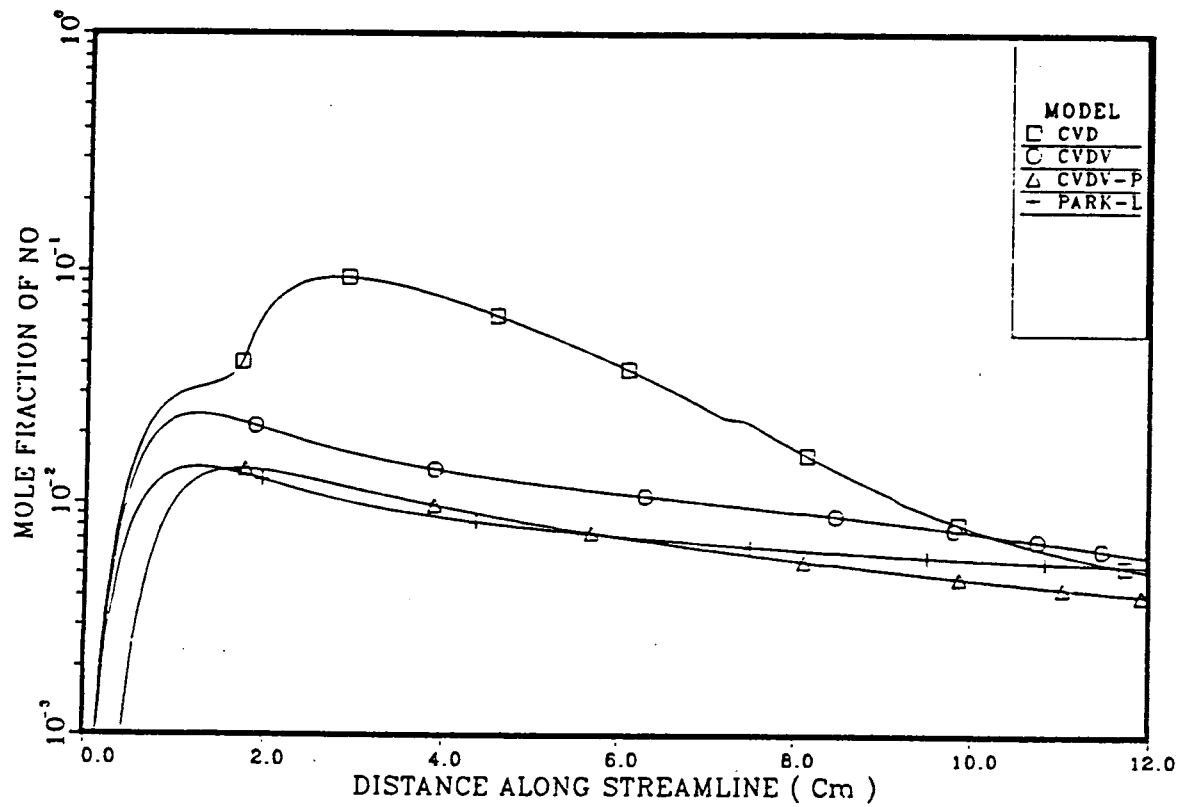


101(a)

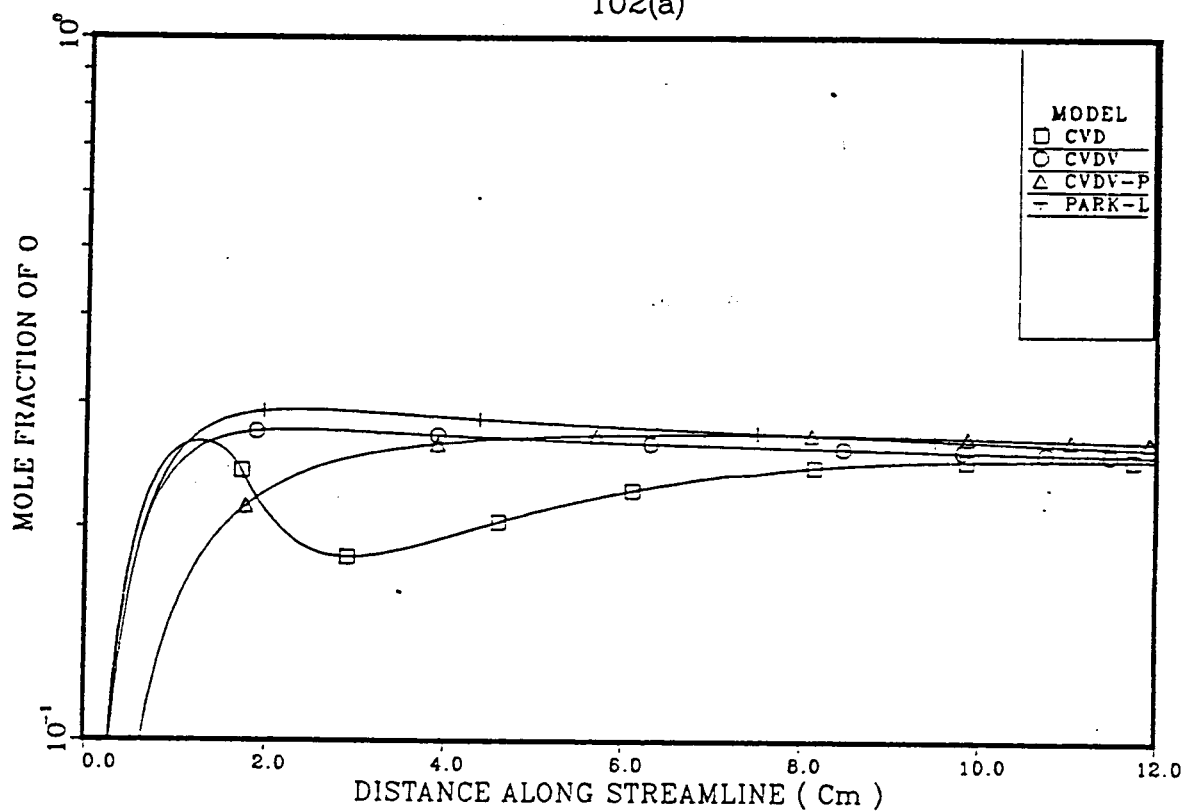


101(b)

FIGURES 101(a),101(b).PROFILES AT V=7.7 Km/s, RR3



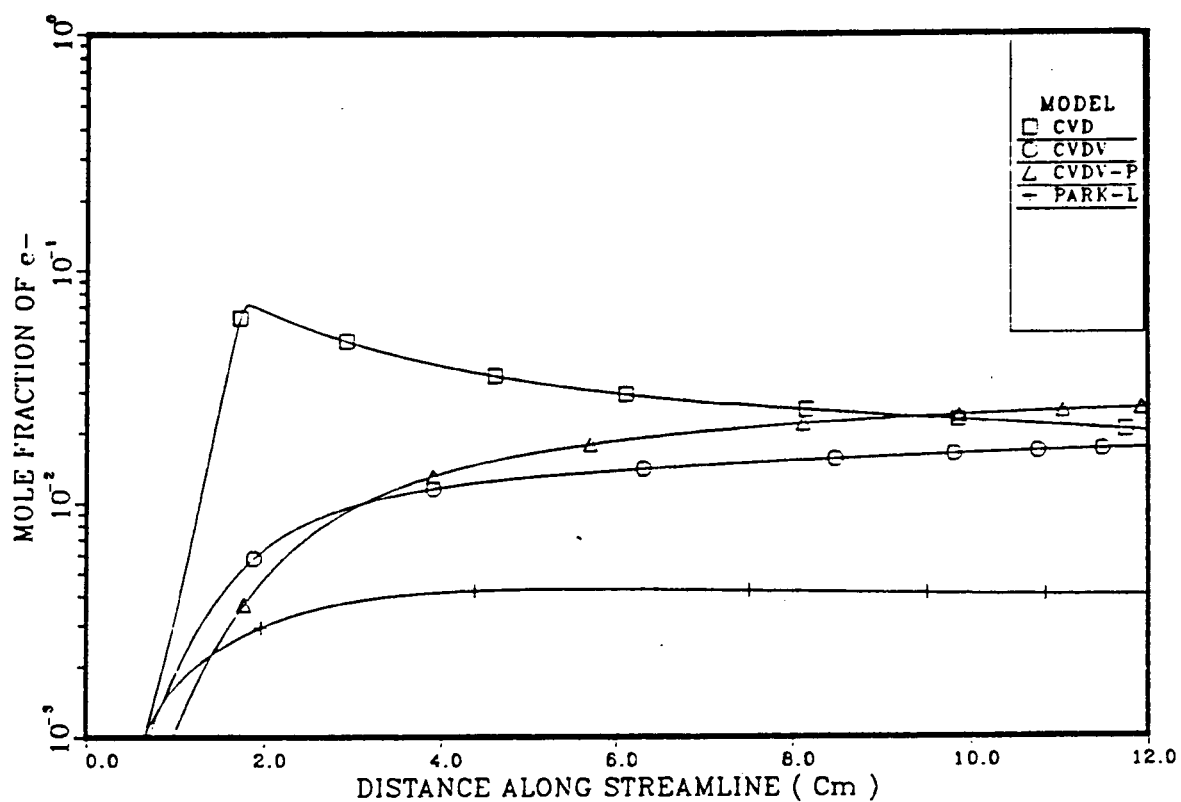
102(a)



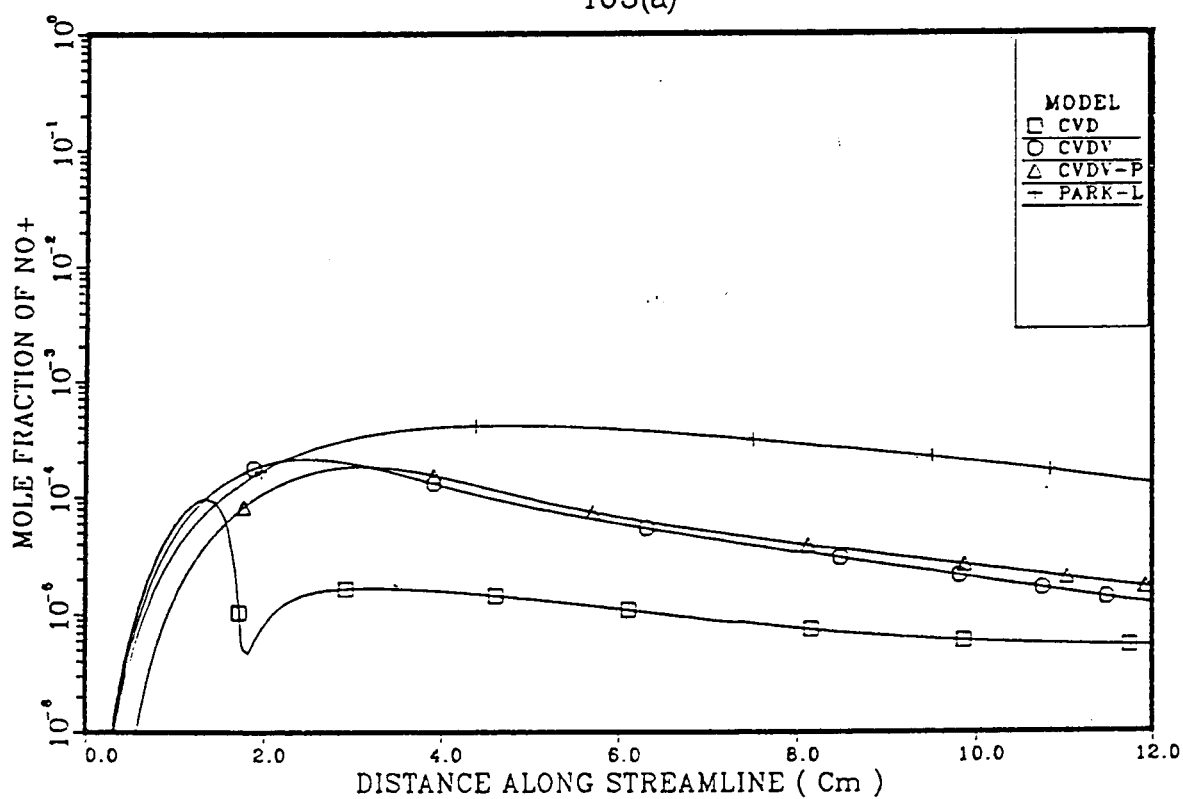
102(b)

FIGURES 102(a),102(b).PROFILES AT V=7.7 Km/s, RR3

ORIGINAL PAGE IS
OF POOR QUALITY

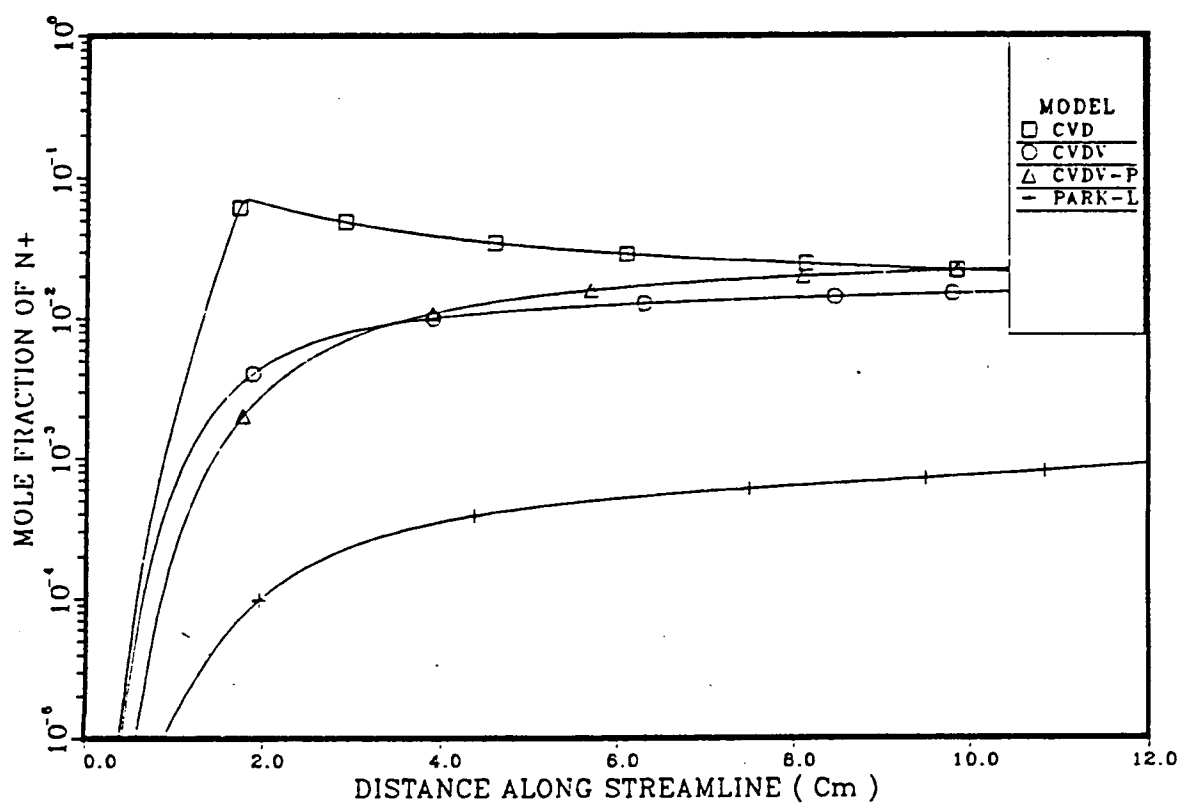


103(a)

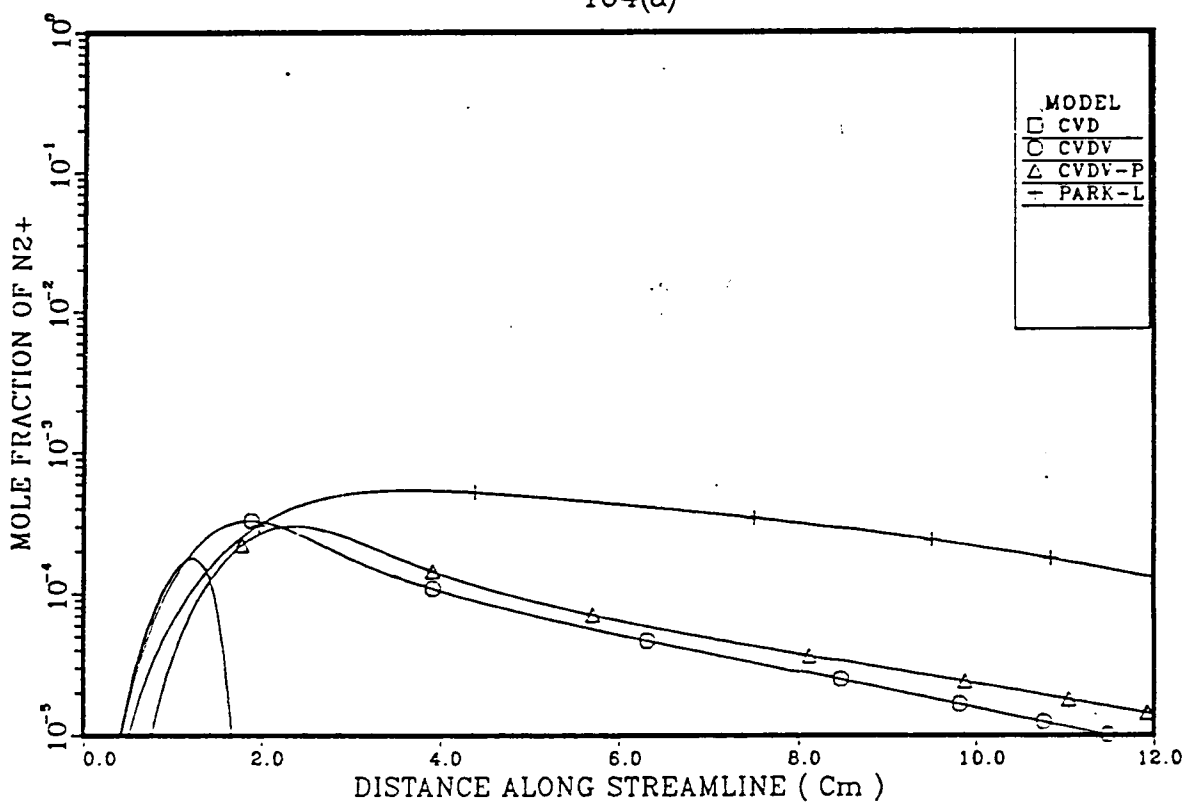


103(b)

FIGURES 103(a),103(b).PROFILES AT V=7.7 Km/s, RR3



104(a)



104(b)

FIGURES 104(a),104(b).PROFILES AT $V=7.7$ Km/s, RR3

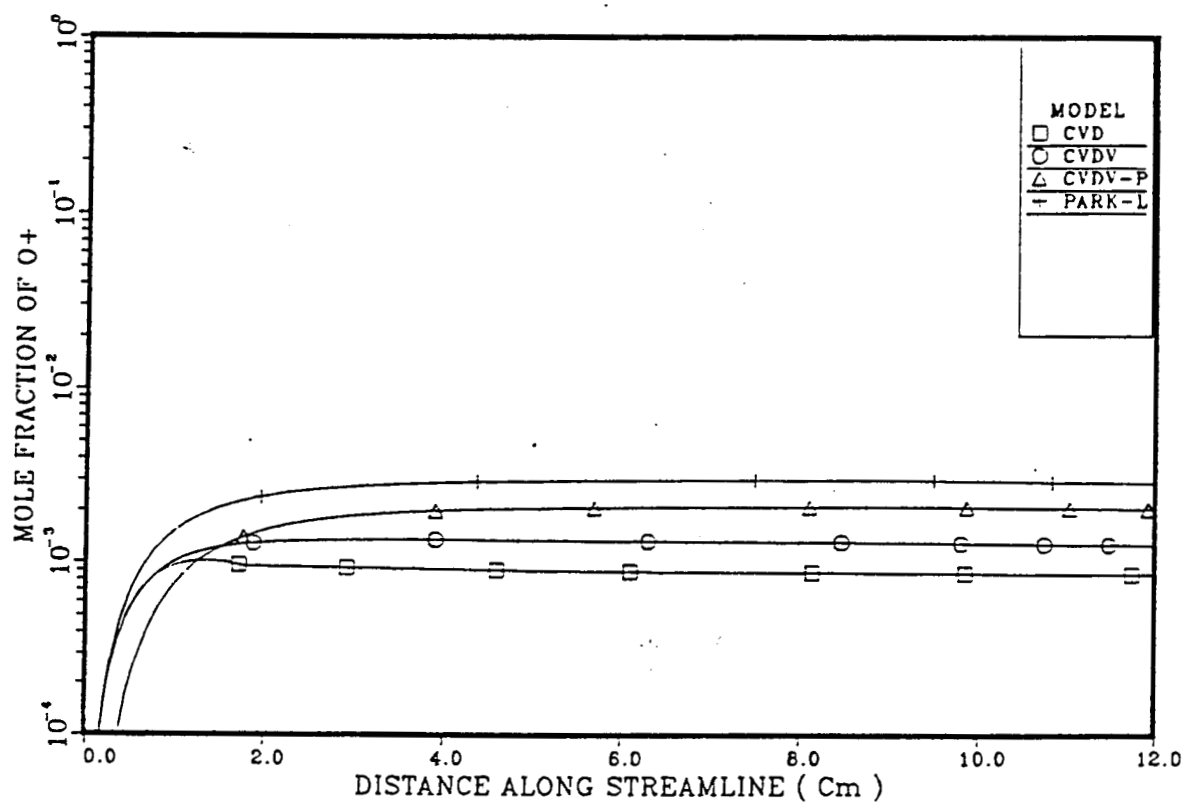
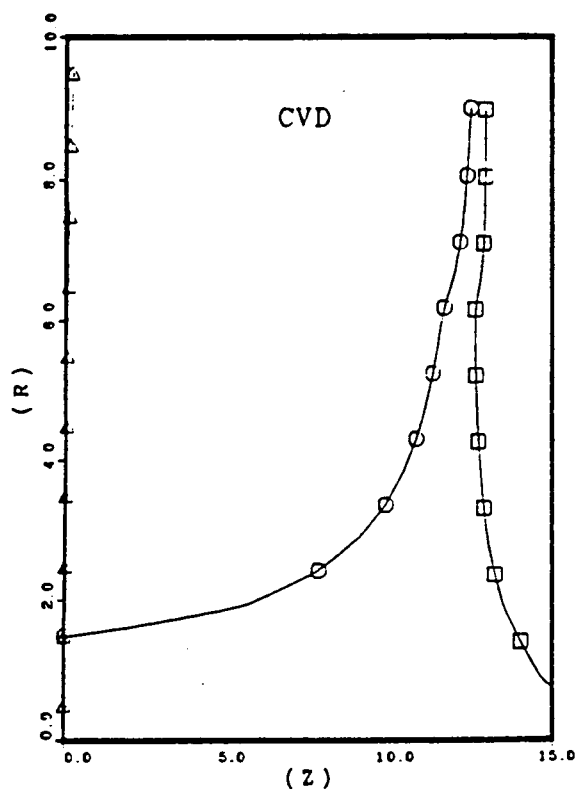
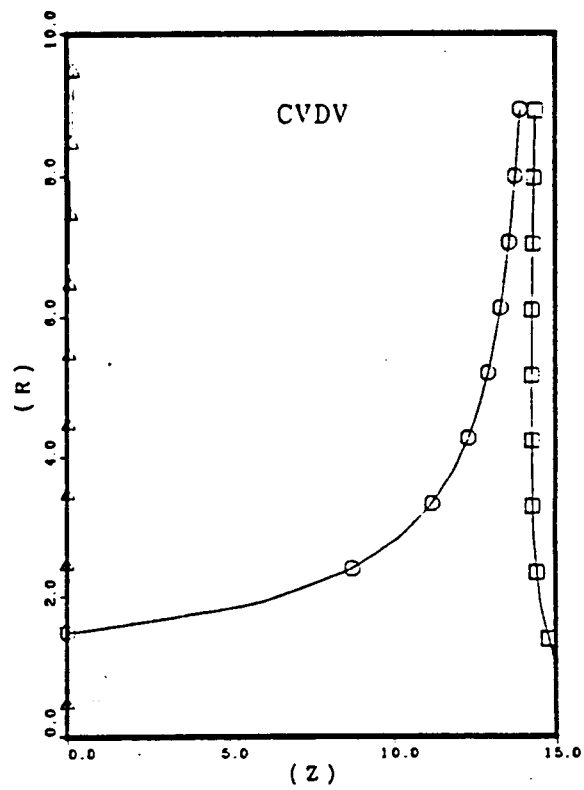


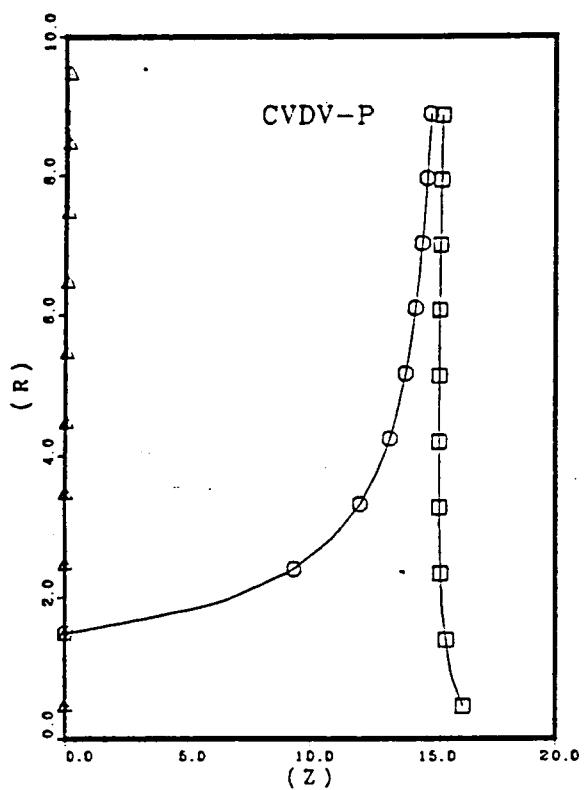
FIGURE 105.PROFILE AT V=7.7 Km/s, RR3



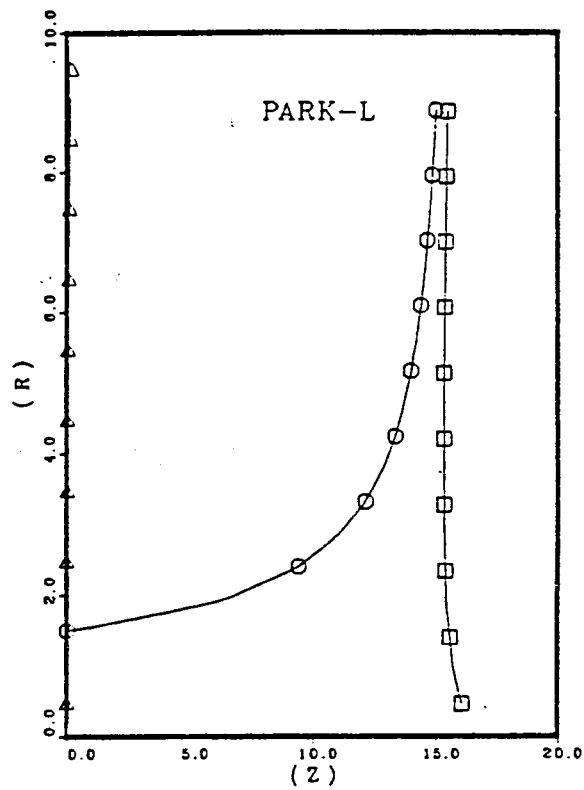
106(a)



106(b)



106(c)

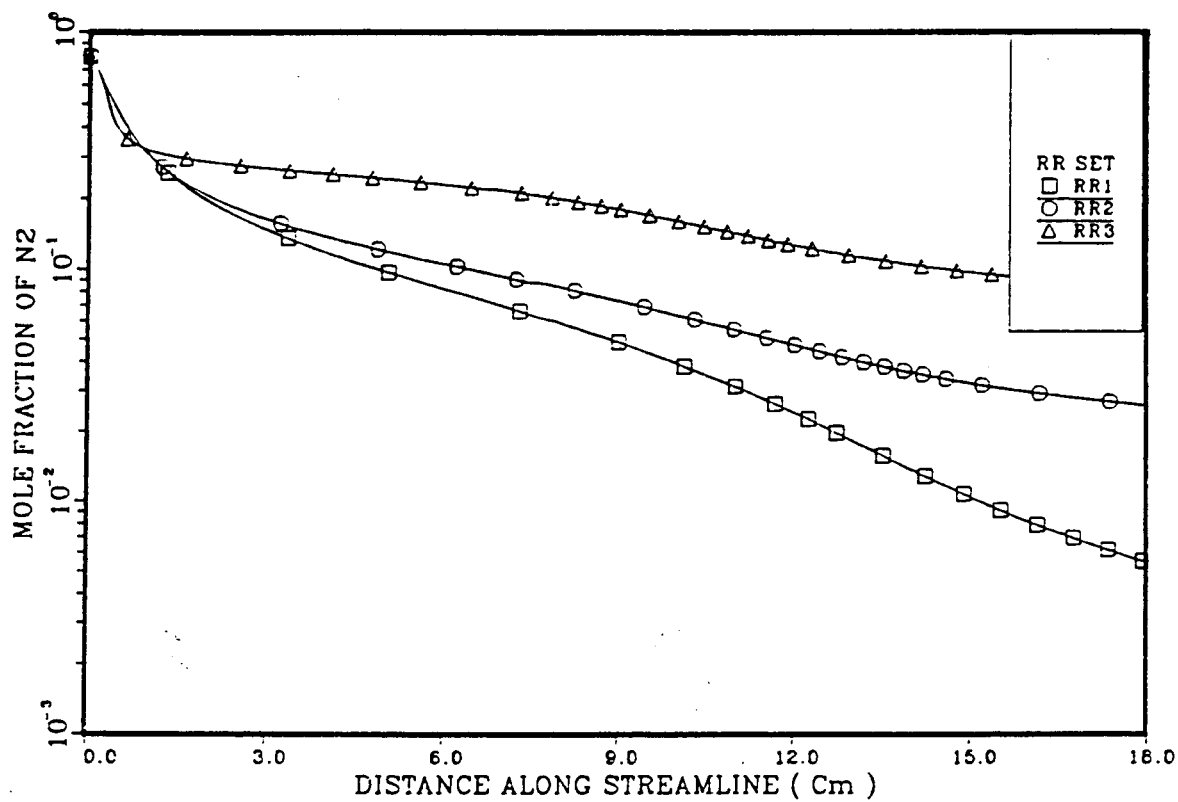


106(d)

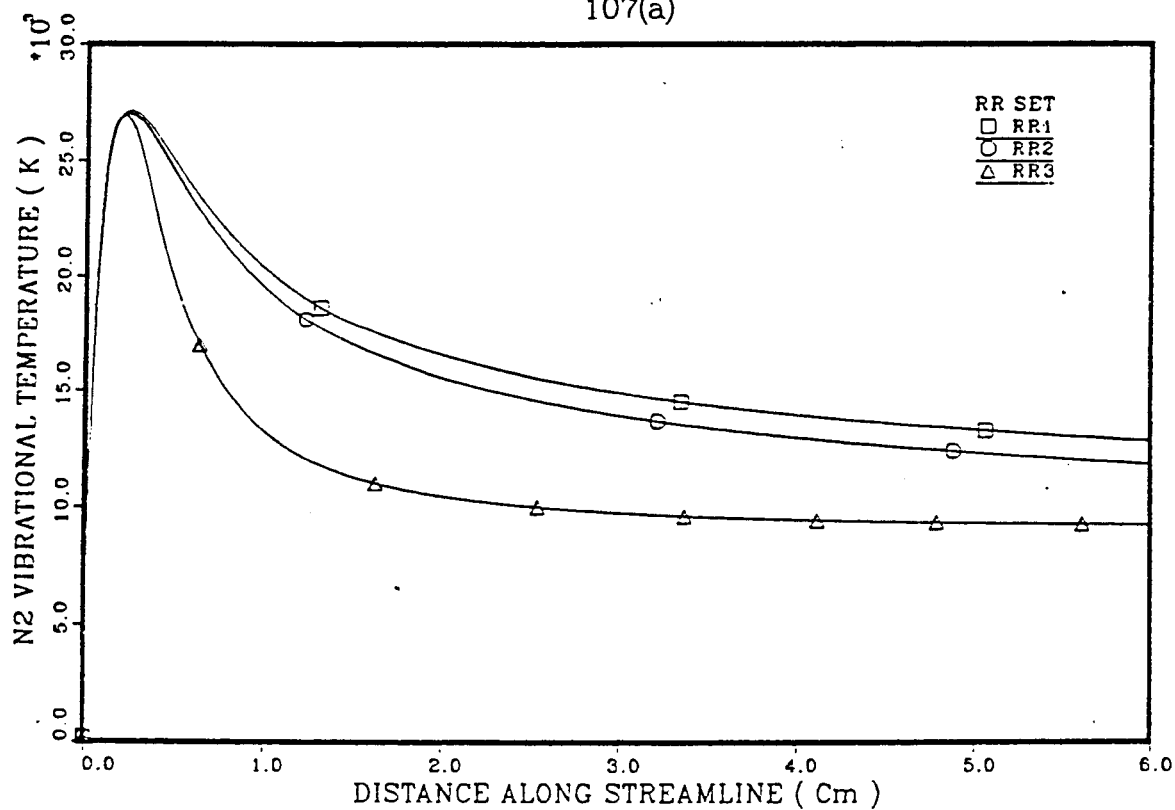
FIGURES 106(a),106(b),106(c),106(d)

.COORD,V=7.7 Km/s, RR3

ORIGINAL PAGE IS
OF POOR QUALITY

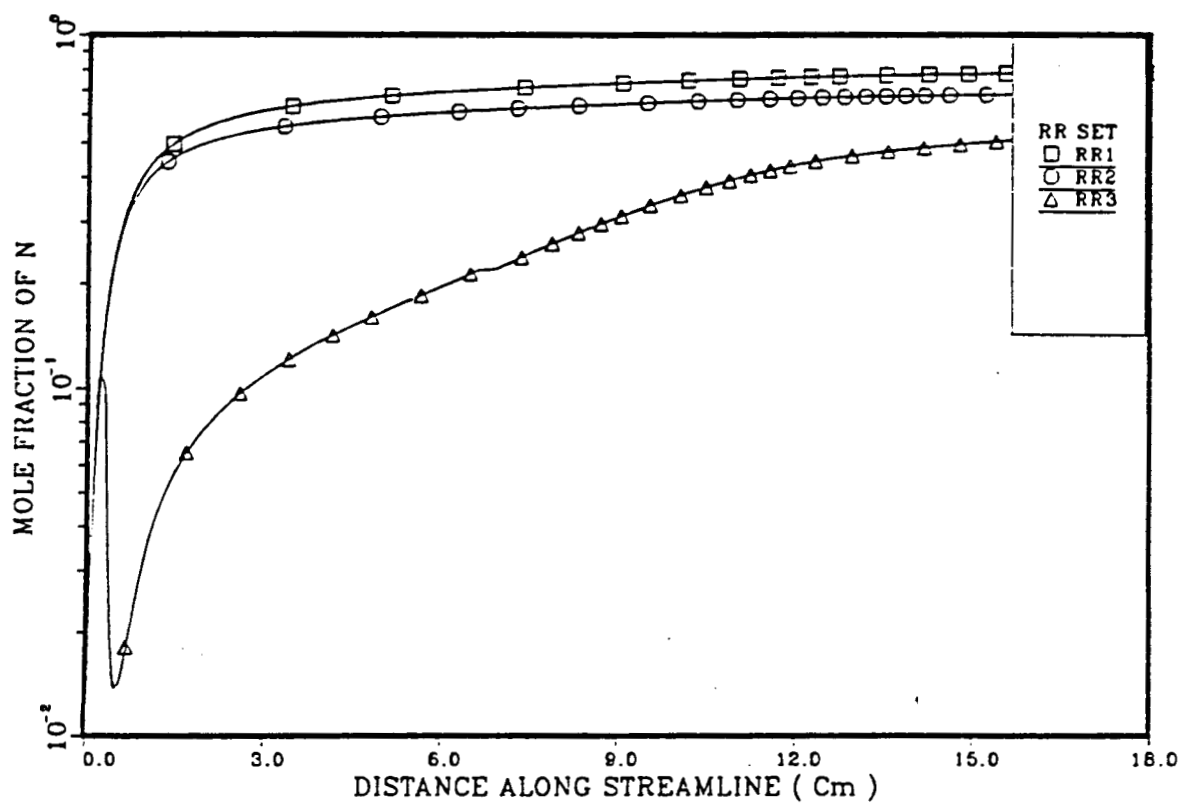


107(a)

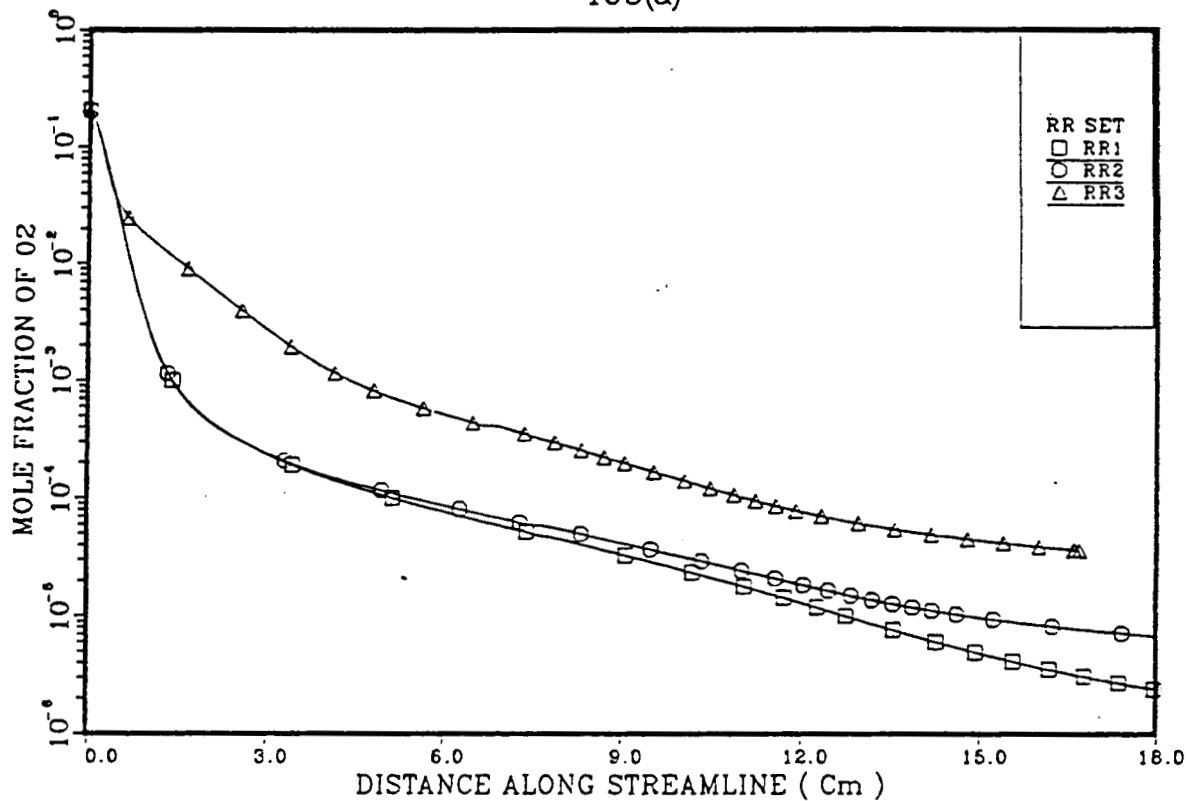


107(b)

FIGURES 107(a),107(b).PROFILES AT V=10 Km/s, CVDV



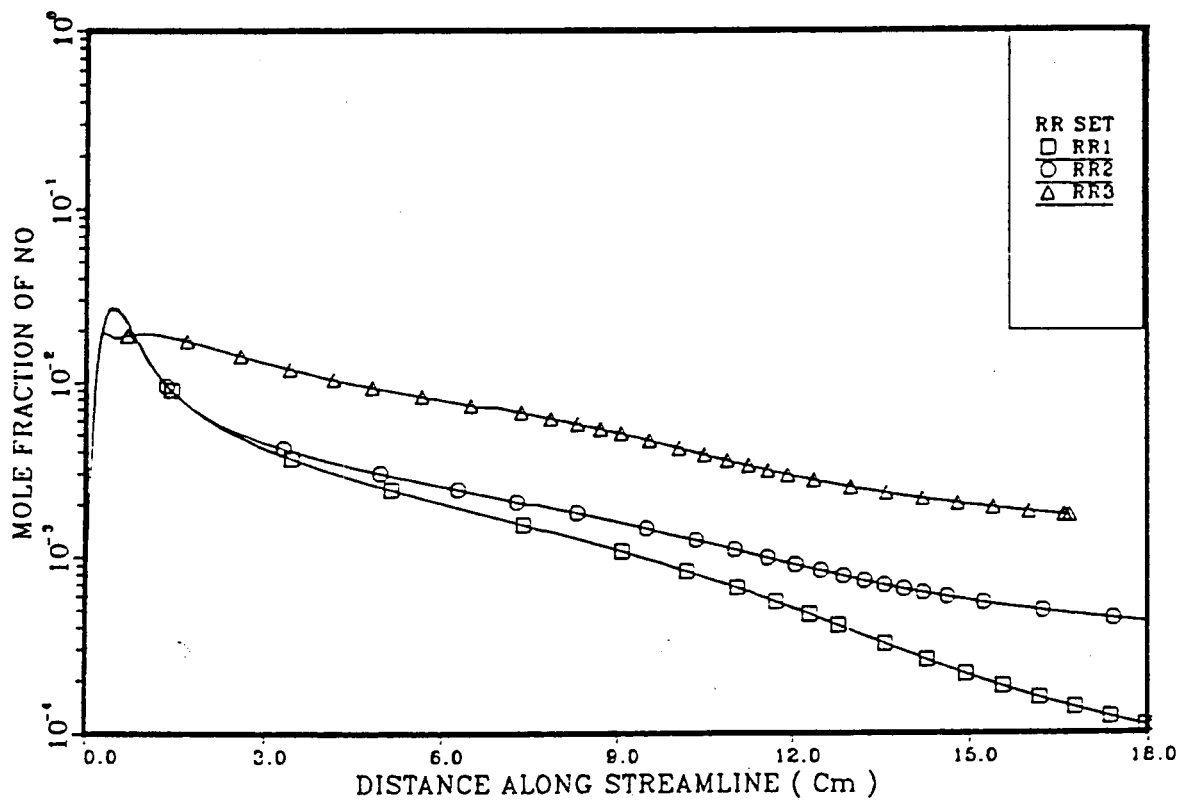
108(a)



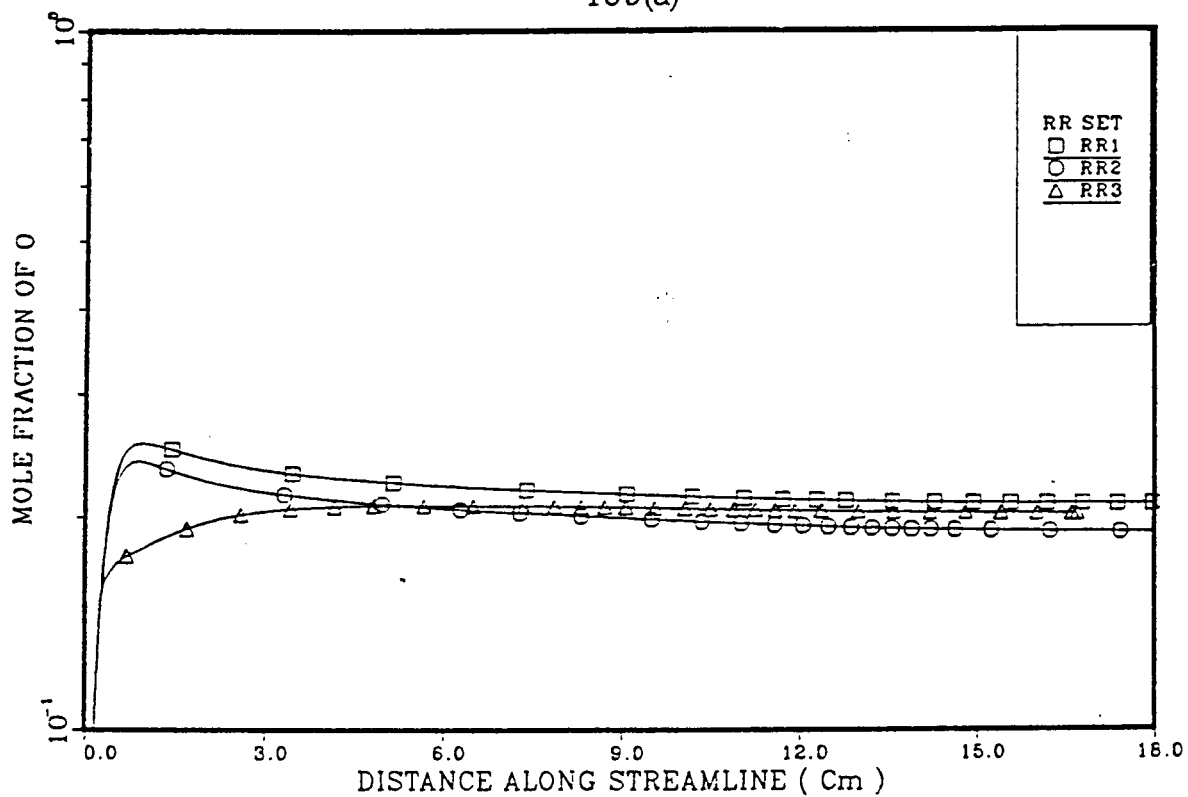
108(b)

FIGURES 108(a),108(b).PROFILES AT V=10 Km/s, CVDV

ORIGINAL PAGE IS
OF POOR QUALITY

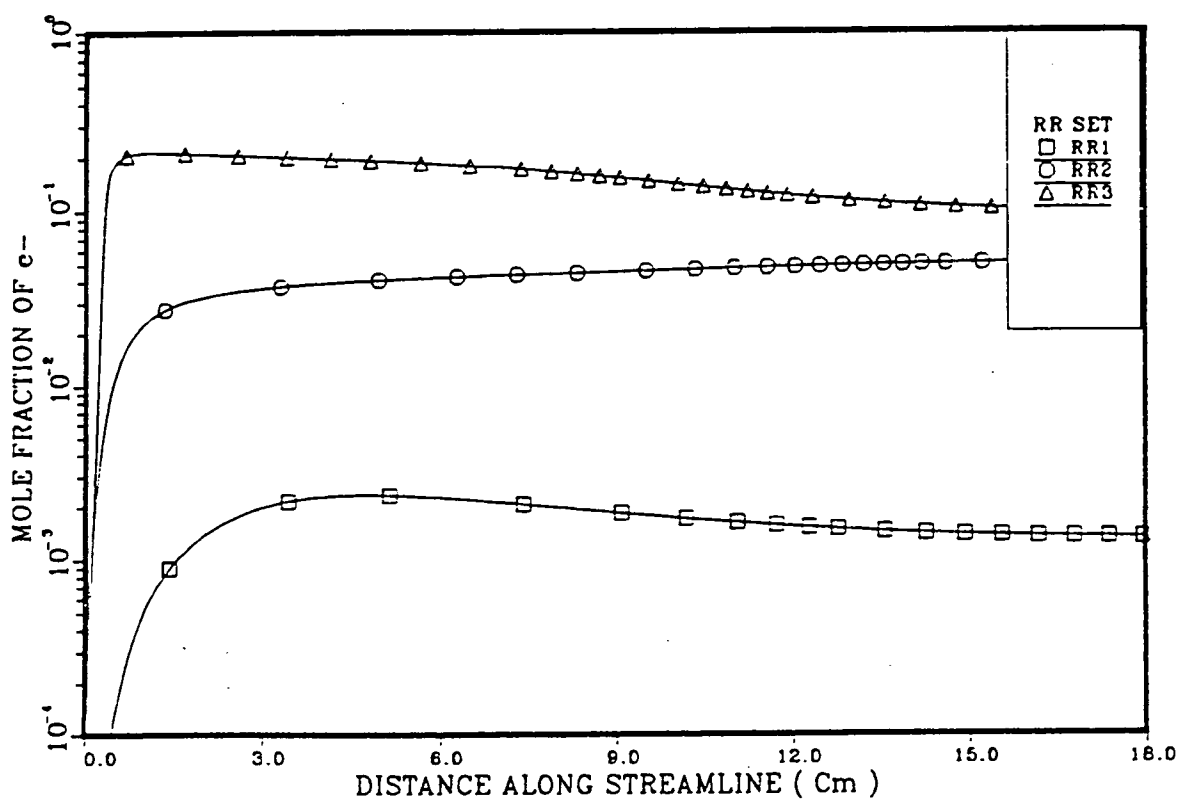


109(a)

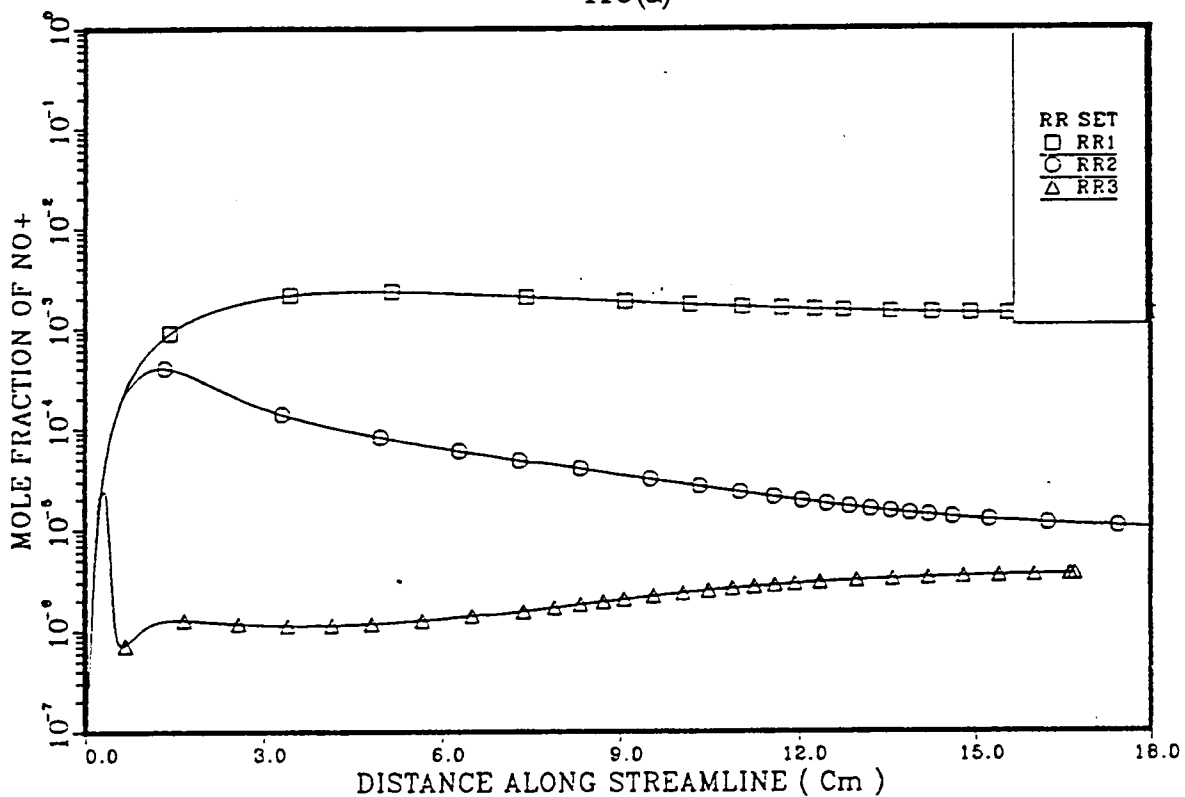


109(b)

FIGURES 109(a),109(b).PROFILES AT V=10 Km/s, CVDV



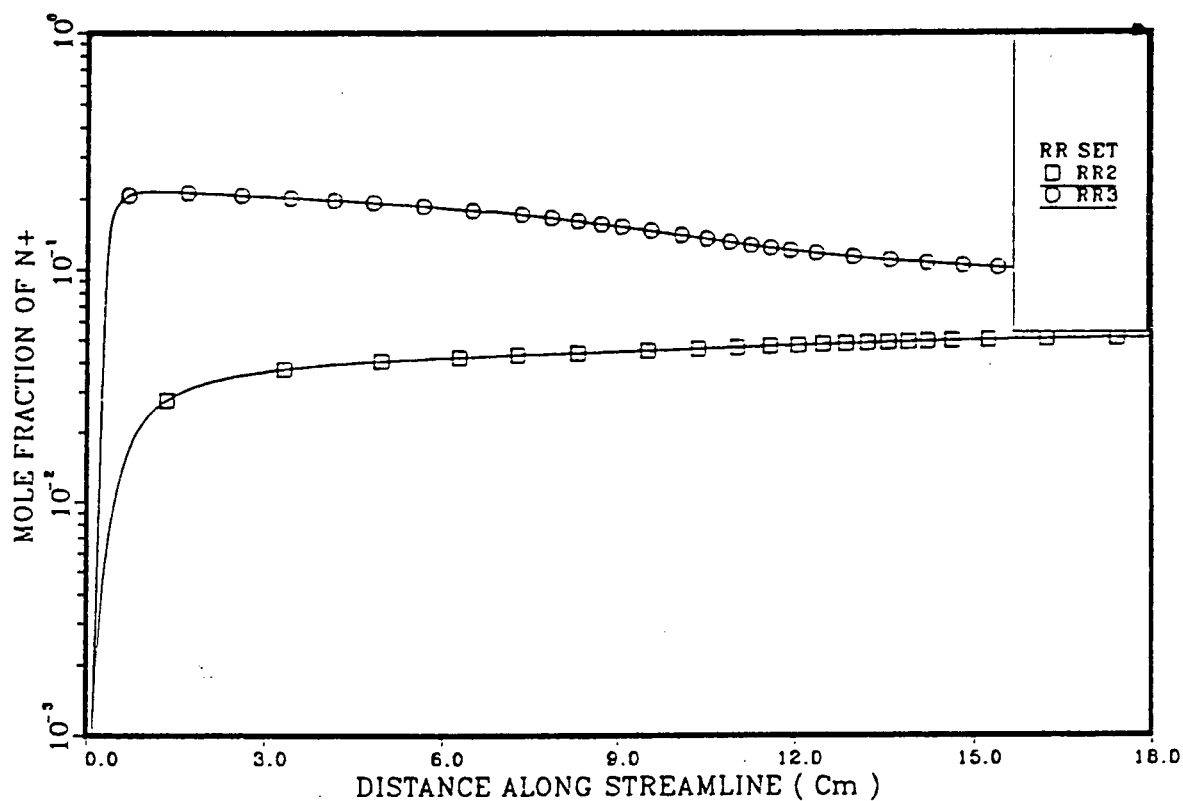
110(a)



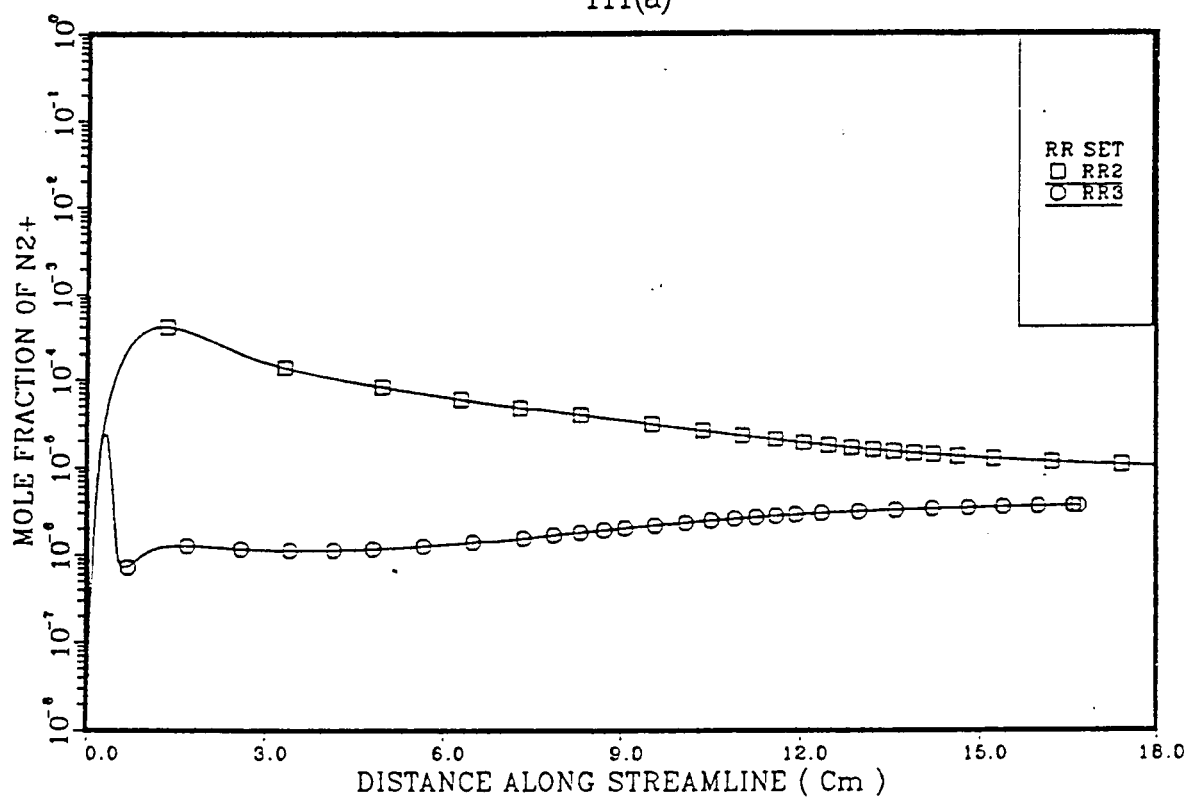
110(b)

FIGURES 110(a),110(b).PROFILES AT $V=10$ Km/s, CVDV

ORIGINAL PAGE IS
OF POOR QUALITY



111(a)



111(b)

FIGURES 111(a),111(b).PROFILES AT $V=10$ Km/s, CVDV

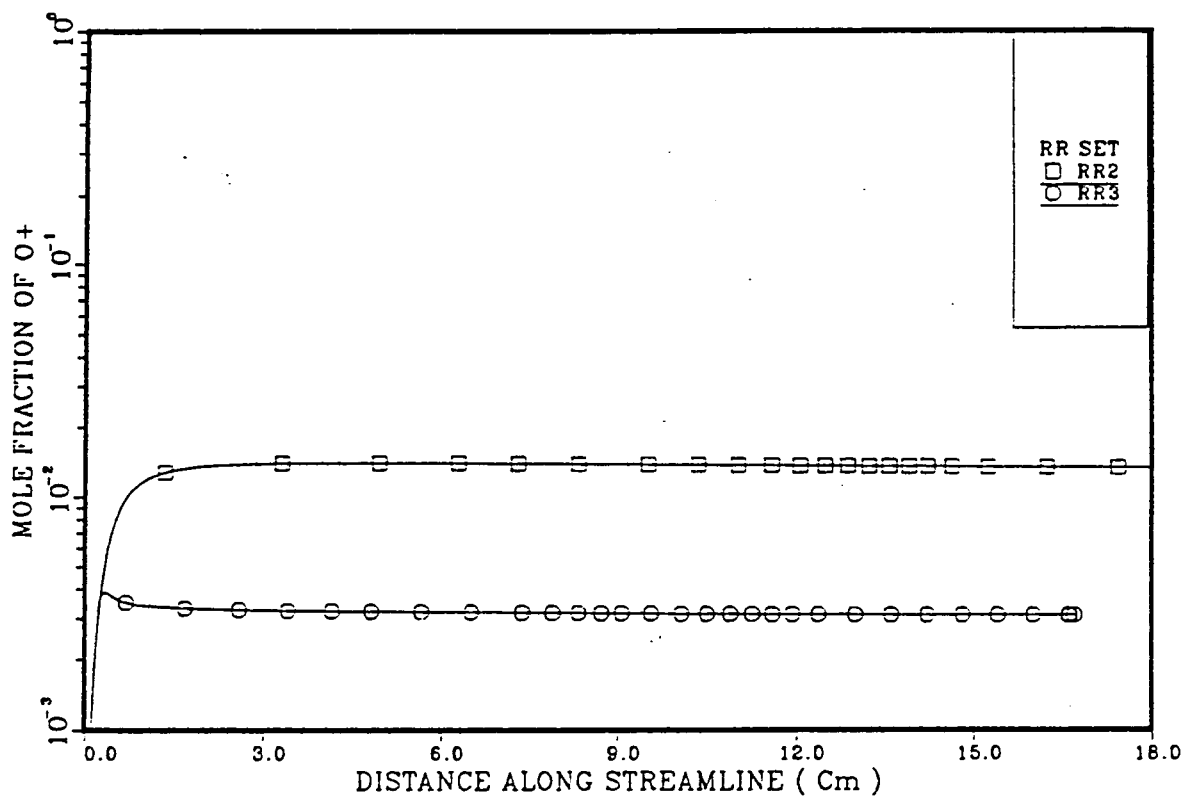
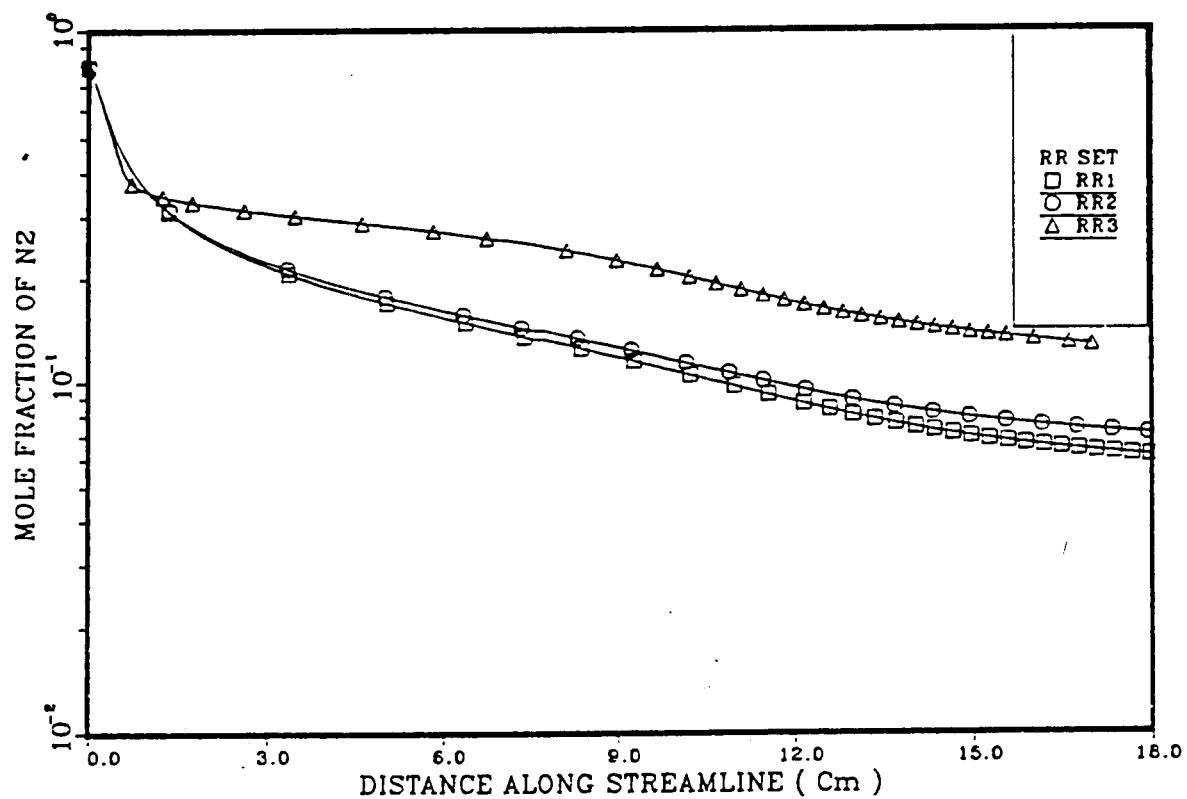
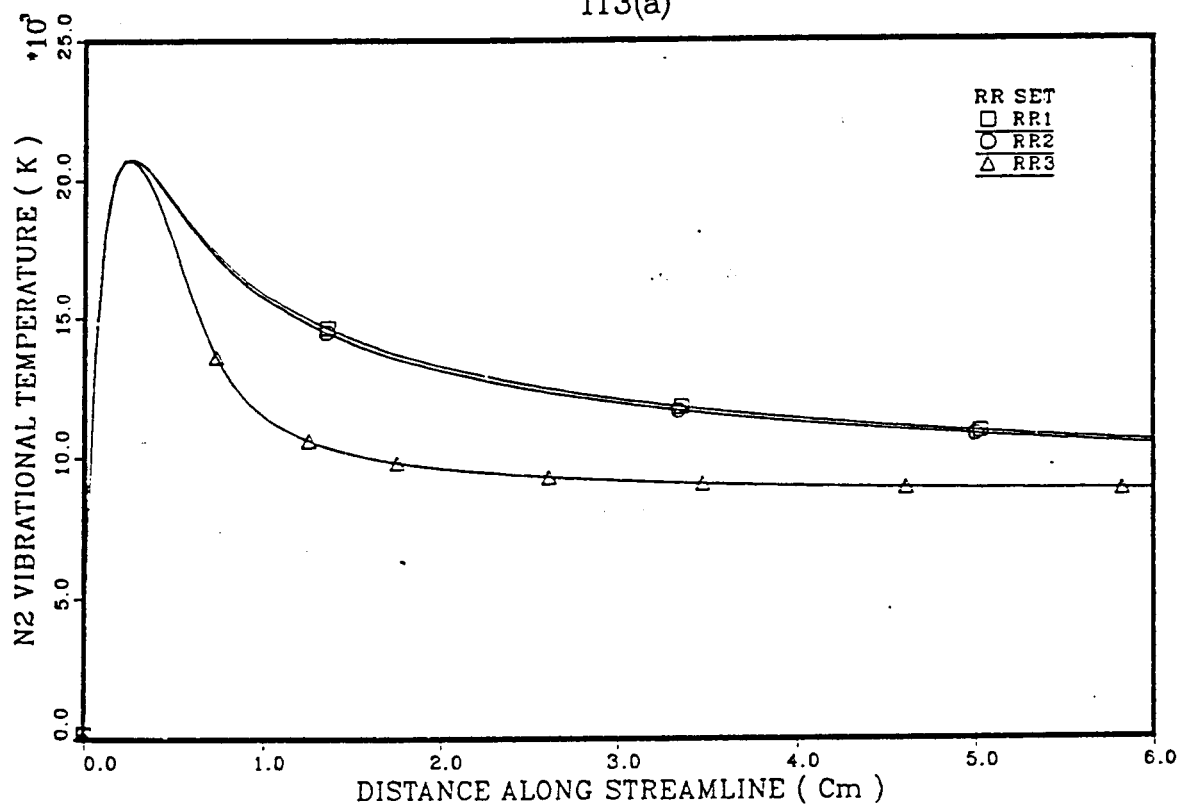


FIGURE 112.PROFILE AT V=10 Km/s, CVDV

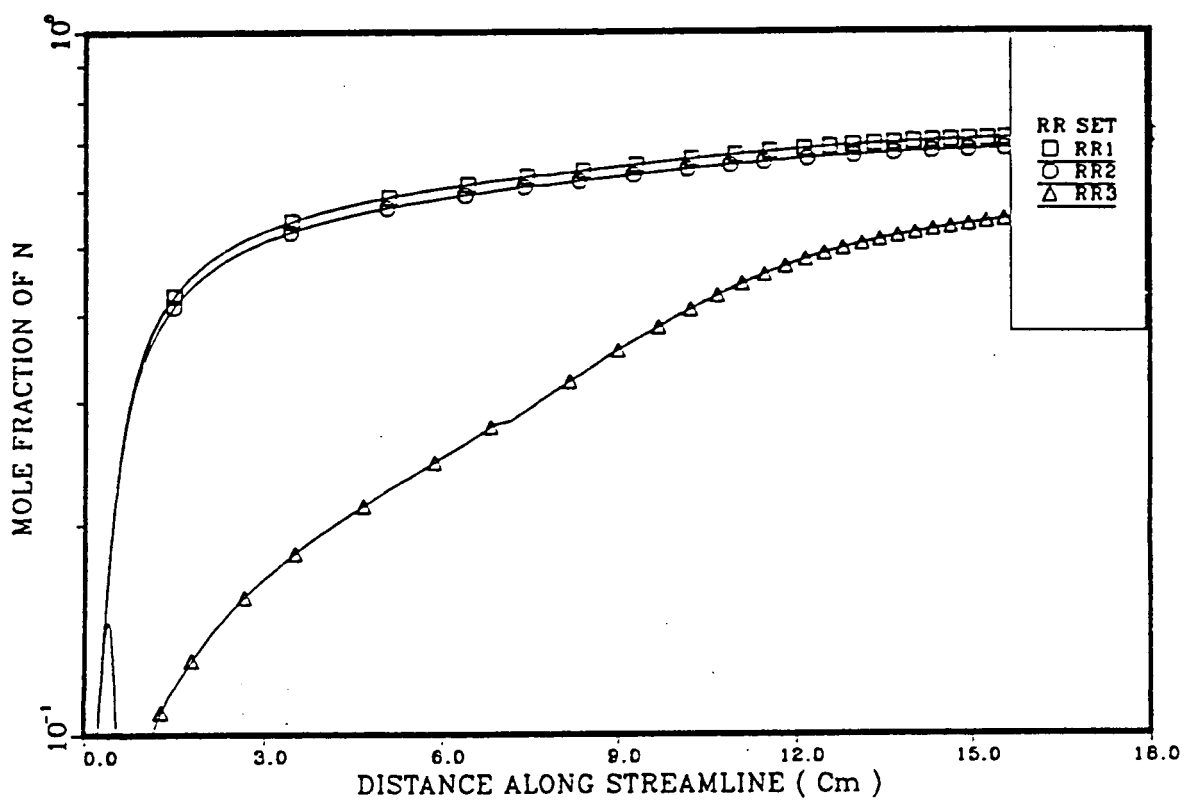


113(a)

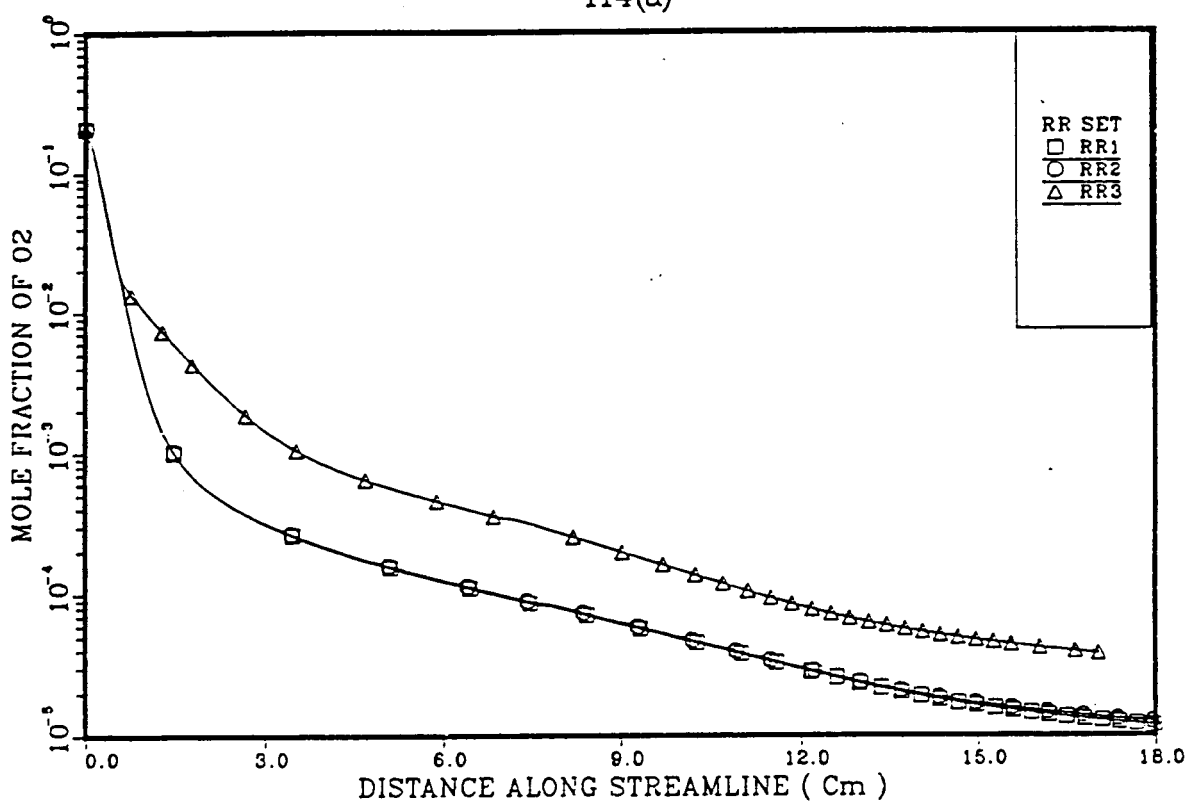


113(b)

FIGURES 113(a),113(b).PROFILES AT V=8.9 Km/s, CVDV

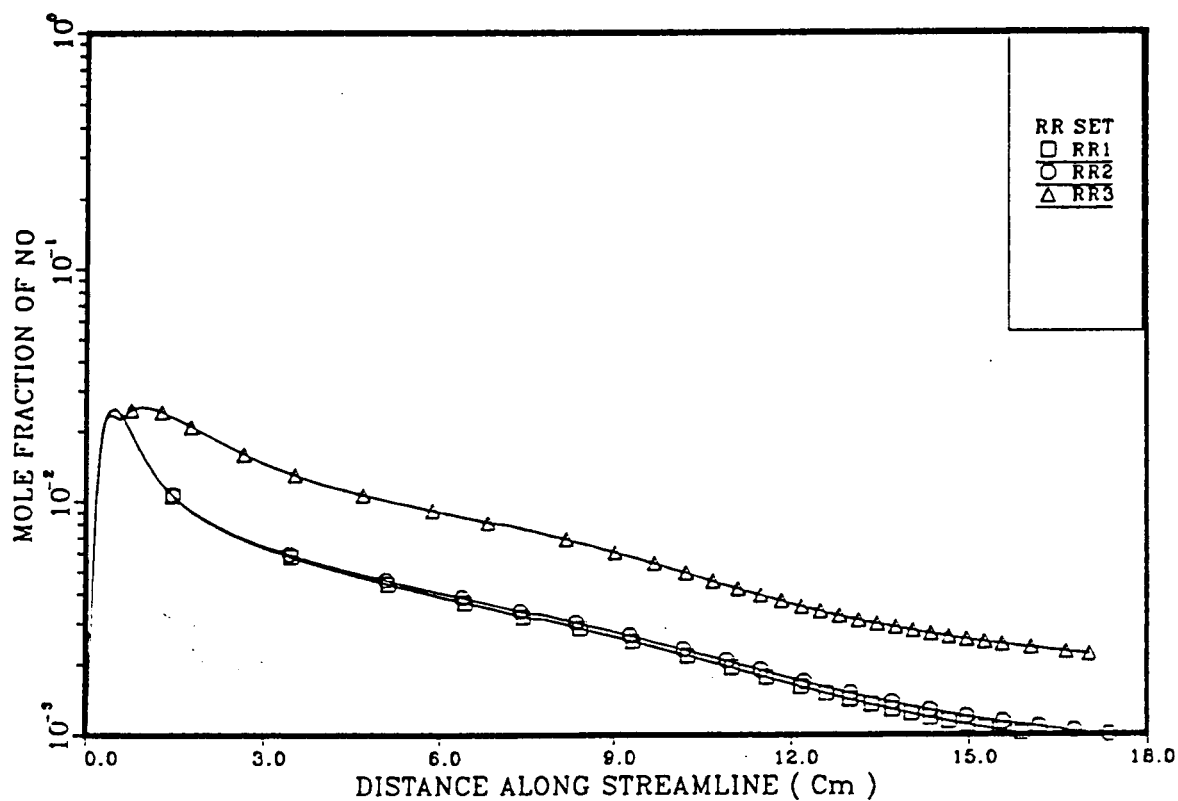


114(a)

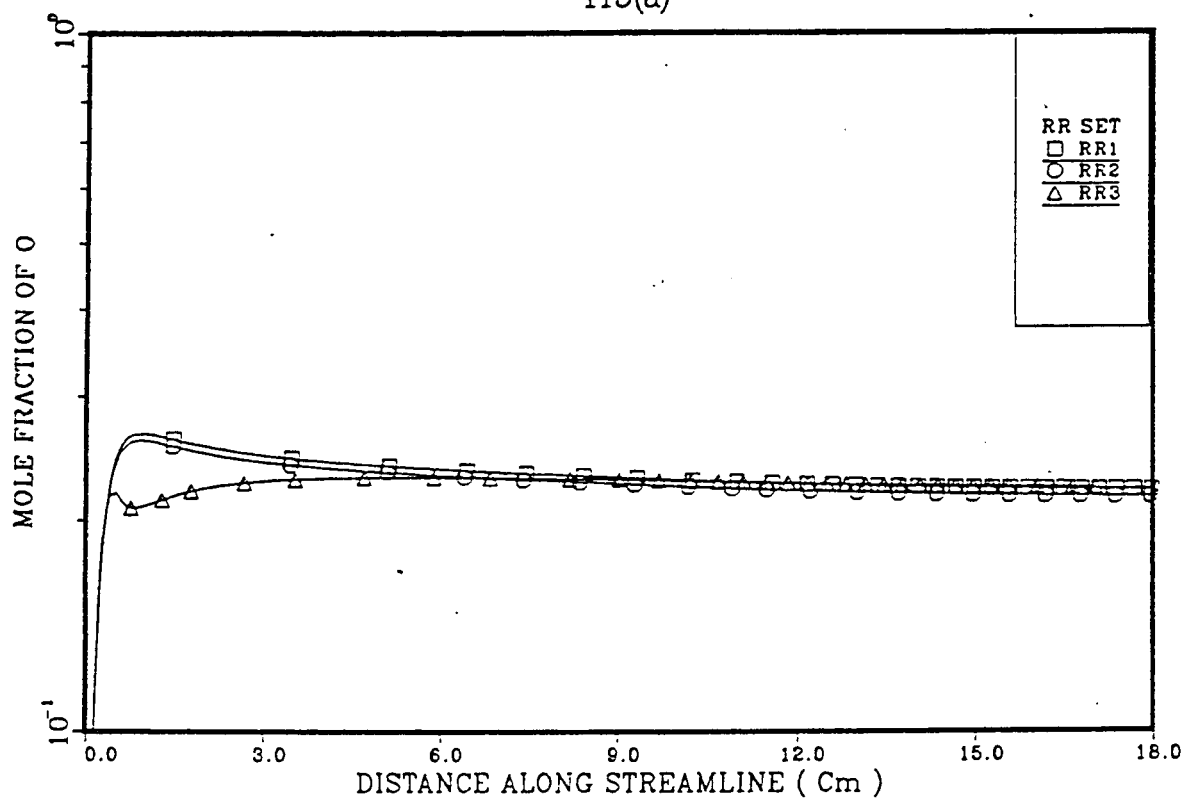


114(b)

FIGURES 114(a), 114(b). PROFILES AT $V=8.9$ Km/s, CVDV

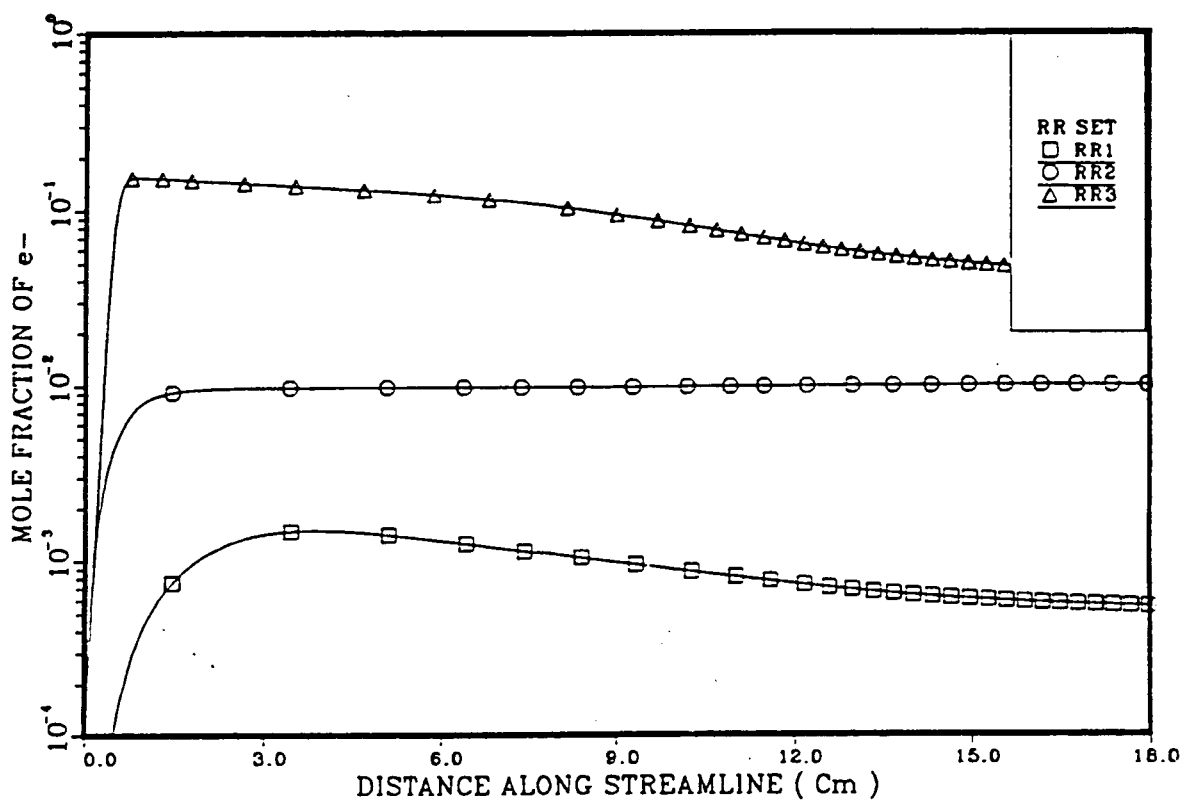


115(a)

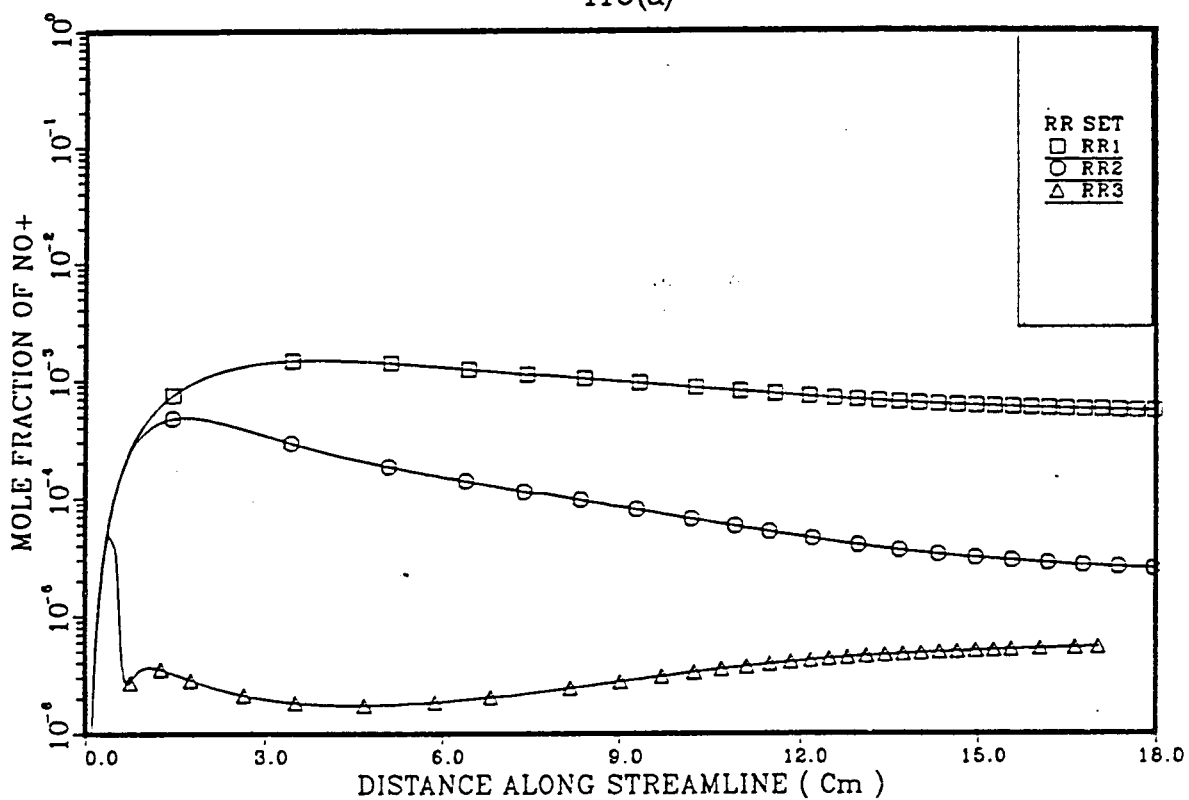


115(b)

FIGURES 115(a),115(b).PROFILES AT $V=8.9$ Km/s, CVDV

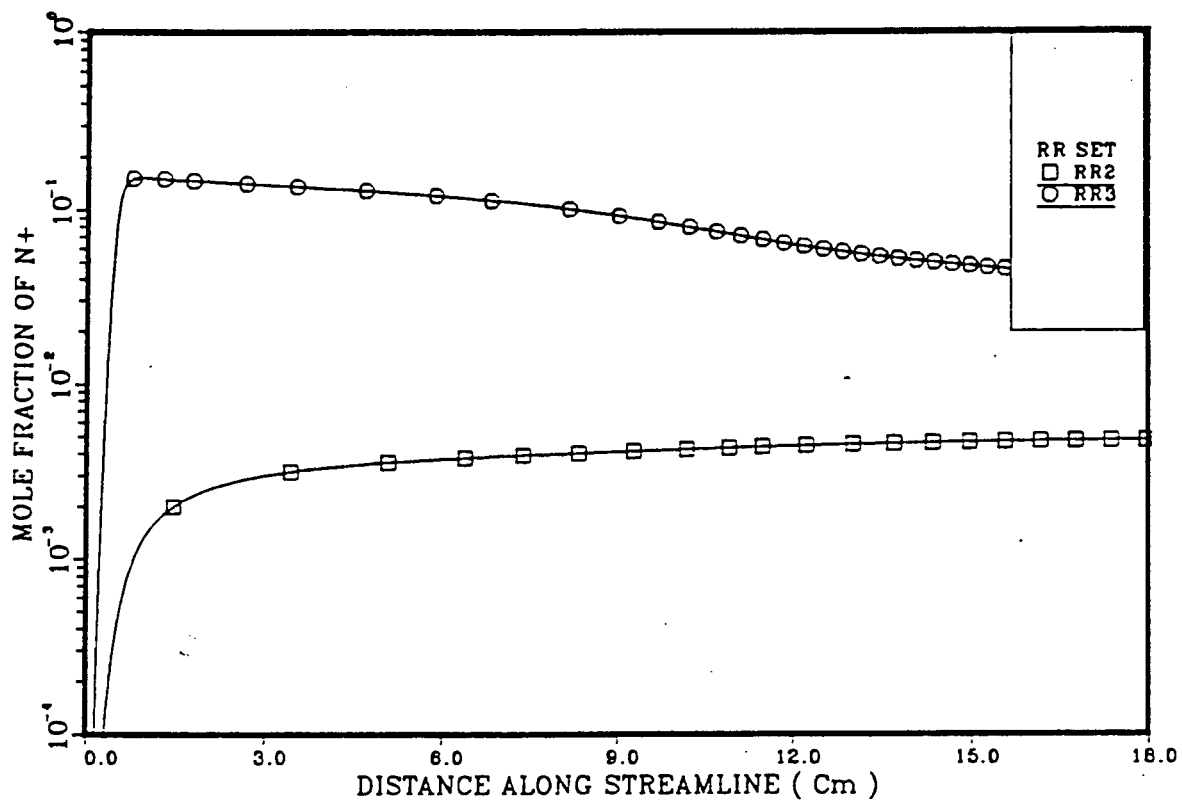


116(a)

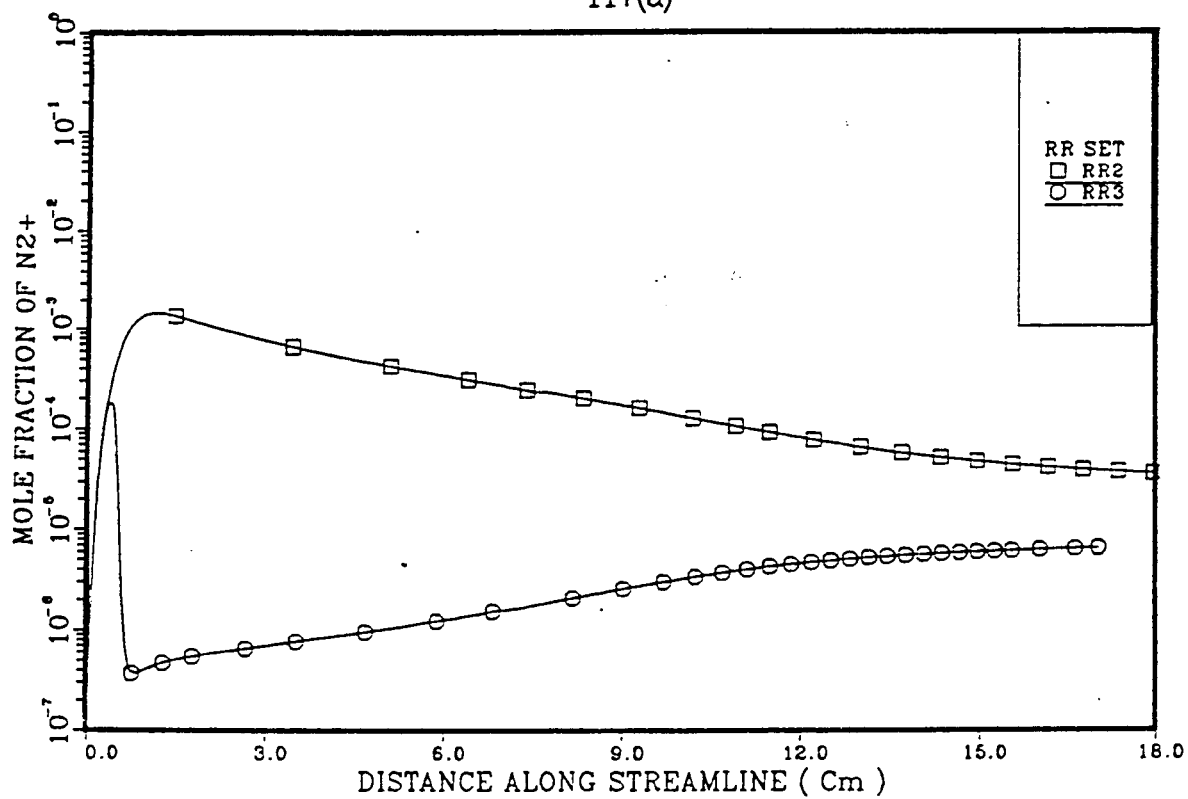


116(b)

FIGURES 116(a),116(b).PROFILES AT $V=8.9$ Km/s, CVDV



117(a)



117(b)

FIGURES 117(a),117(b).PROFILES AT $V=8.9$ Km/s, CVDV

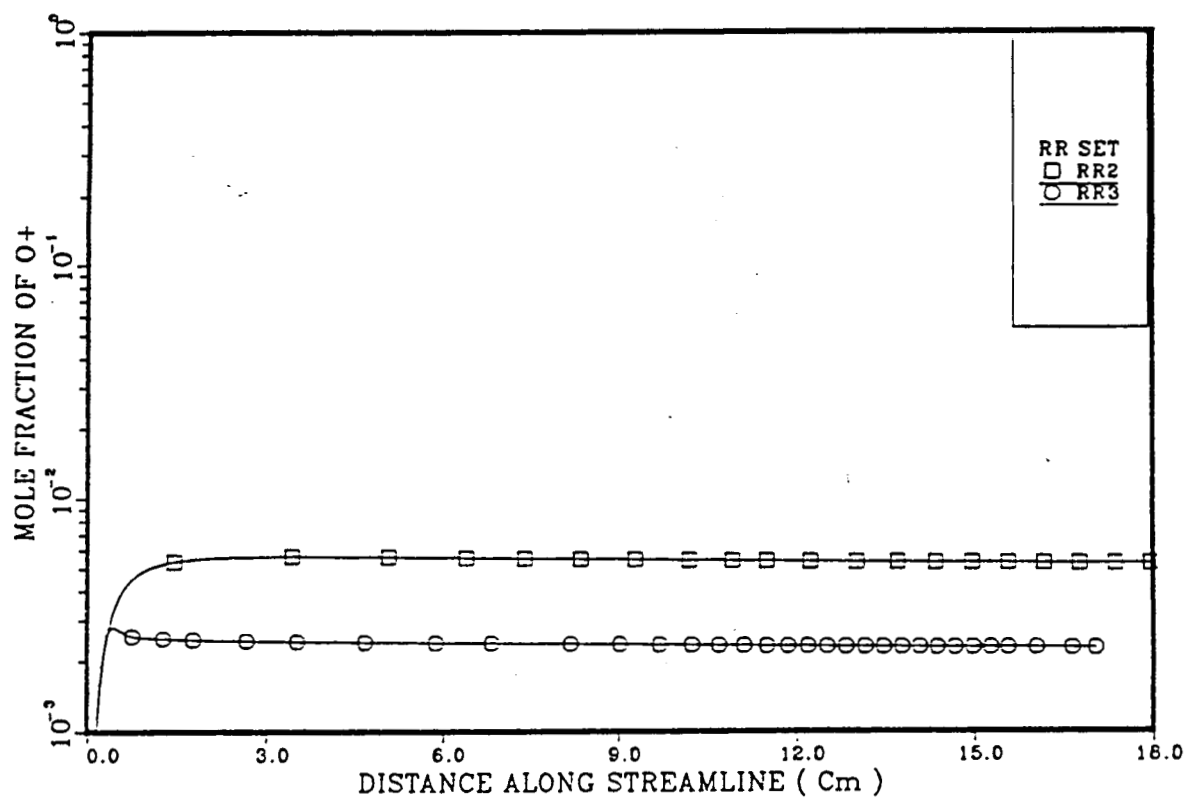
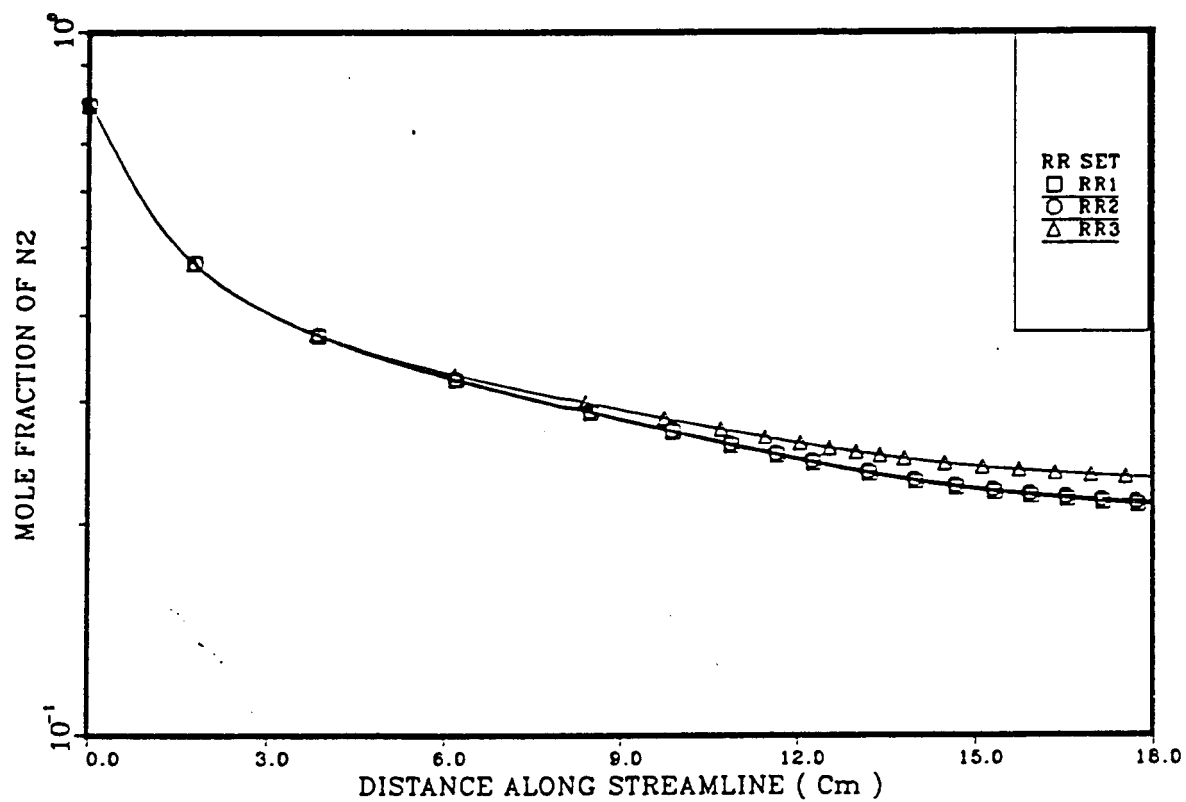
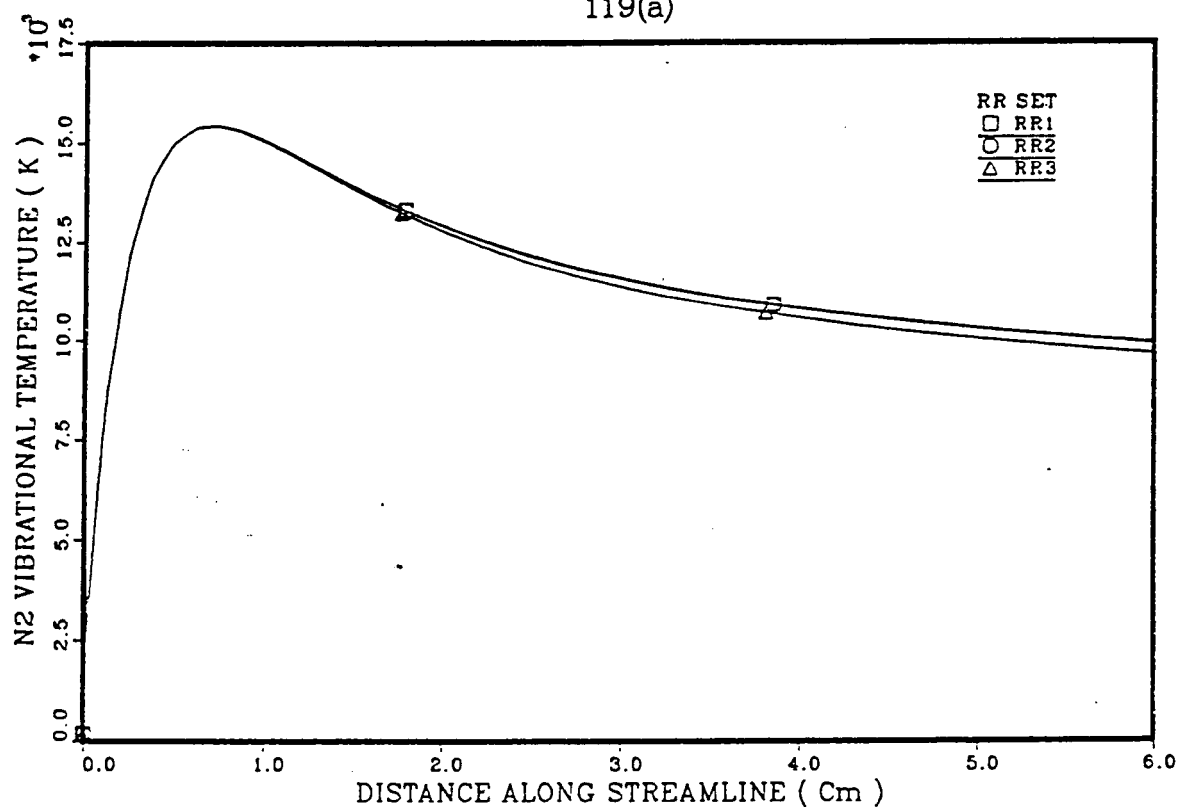


FIGURE 118.PROFILE AT V=8.9 Km/s, CVDV

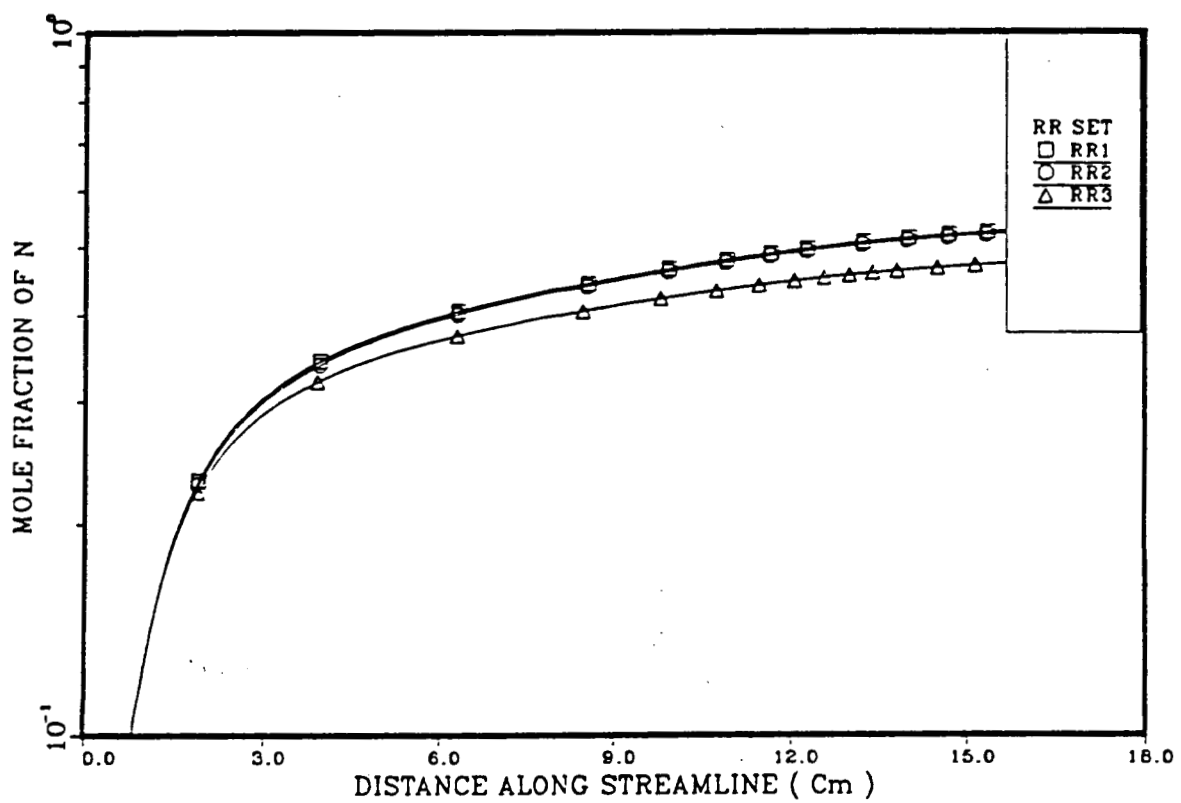


119(a)

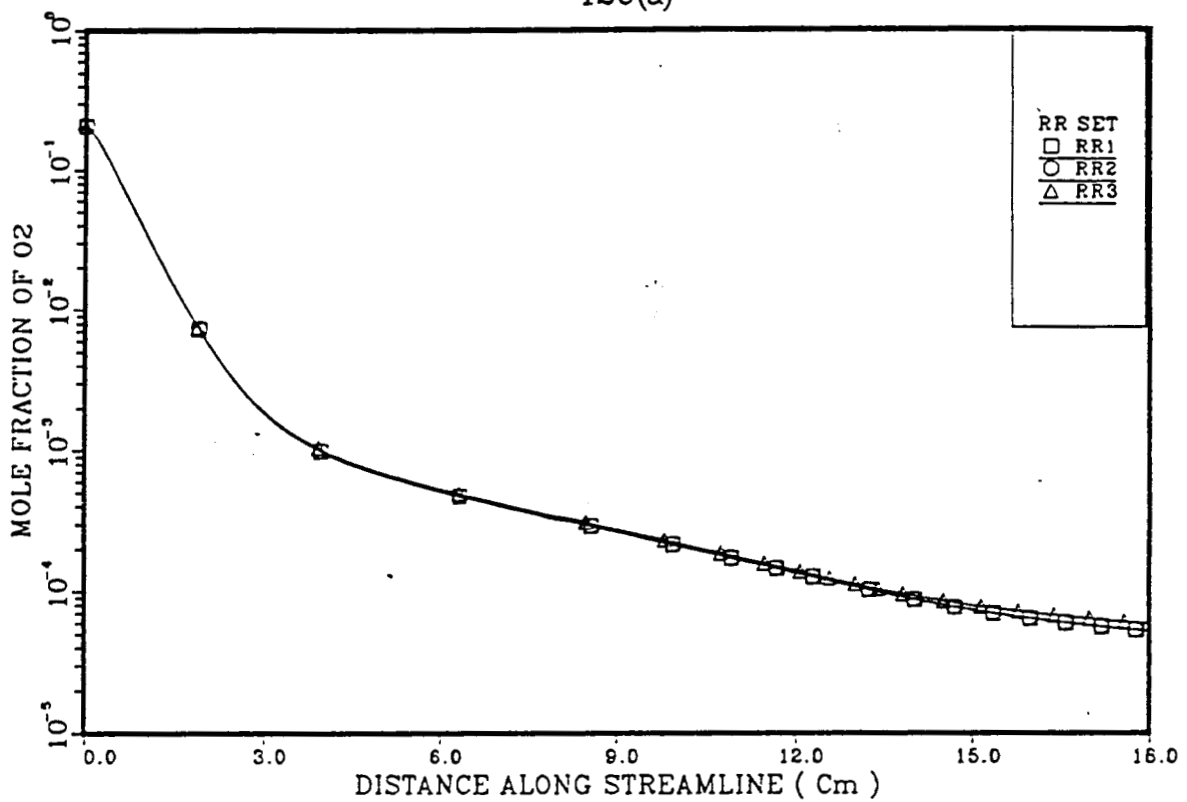


119(b)

FIGURES 119(a),119(b).PROFILES AT V=7.7 Km/s, CVDV

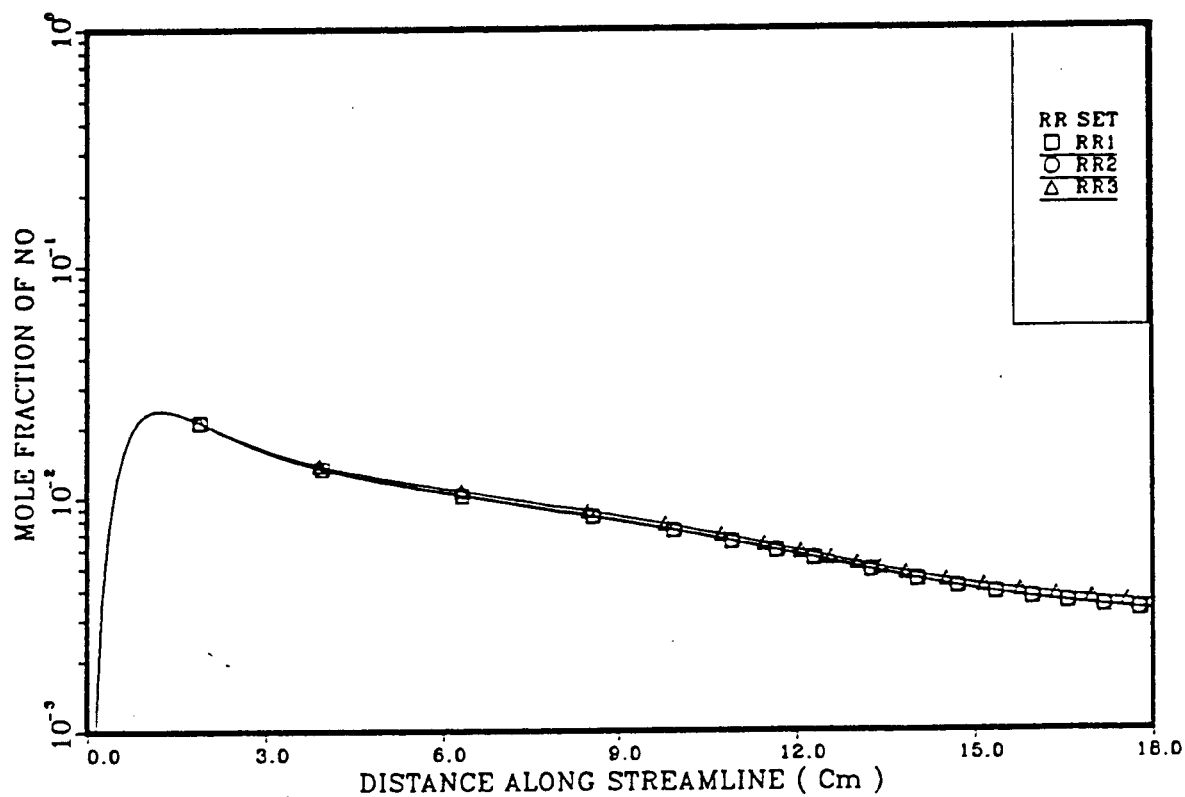


120(a)

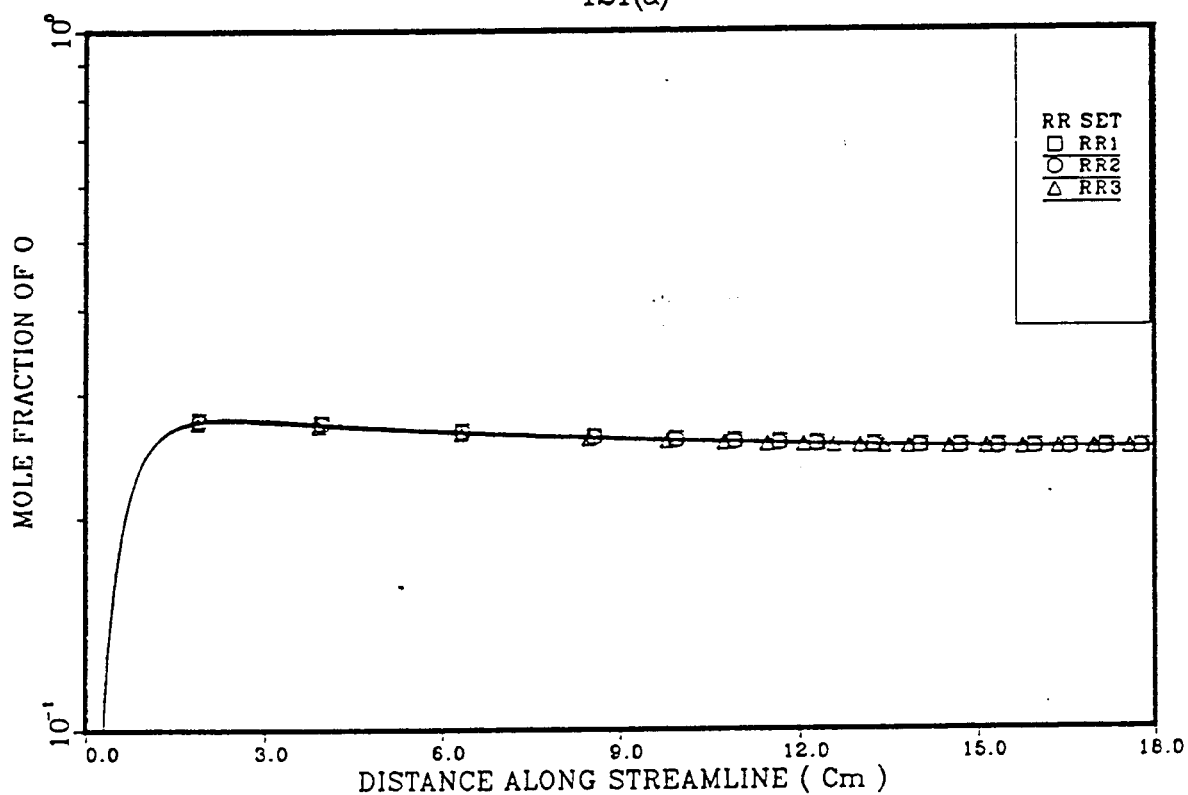


120(b)

FIGURES 120(a),120(b).PROFILES AT V=7.7 Km/s, CVDV

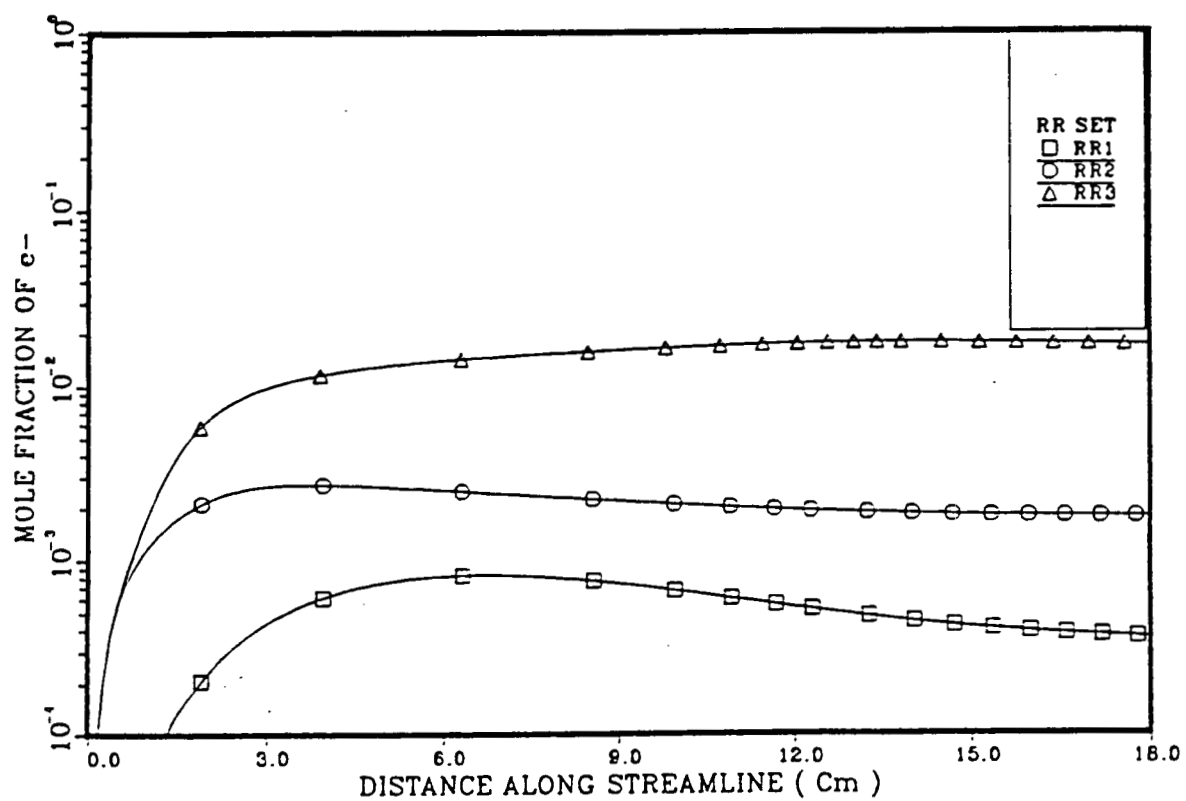


121(a)

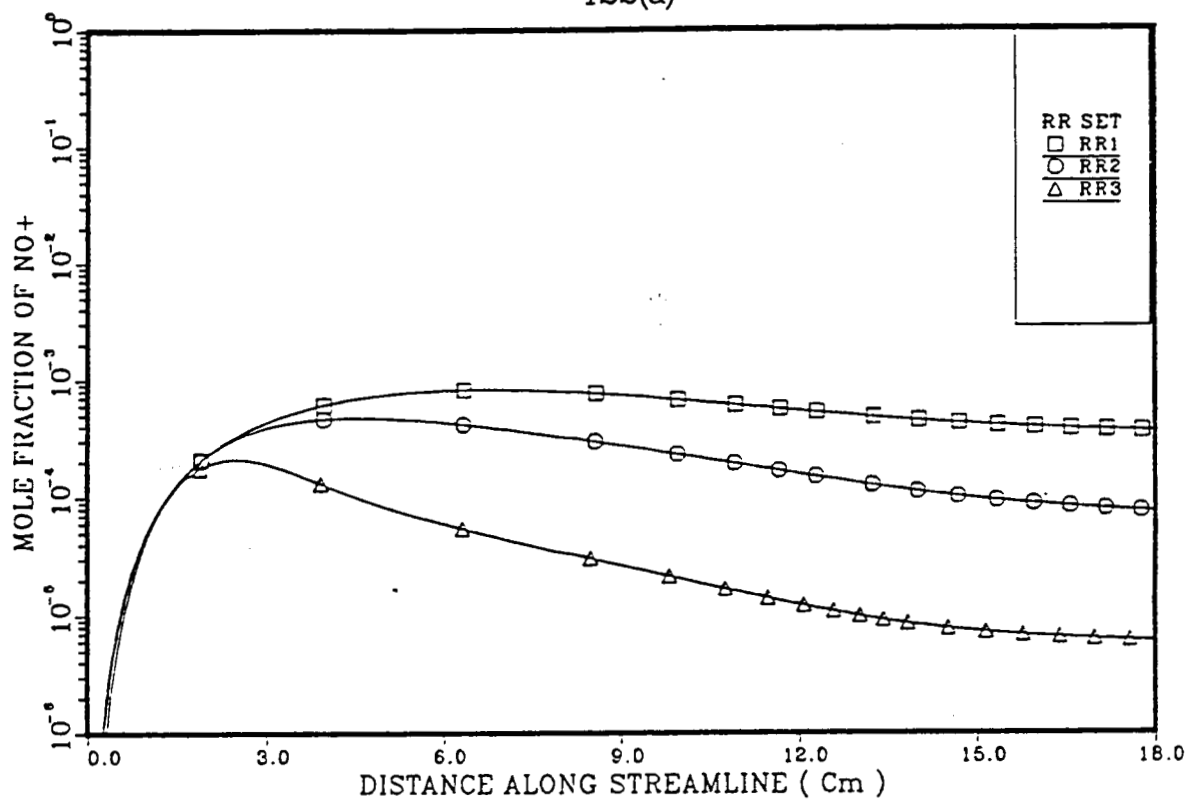


121(b)

FIGURES 121(a),121(b).PROFILES AT V=7.7 Km/s, CVDV

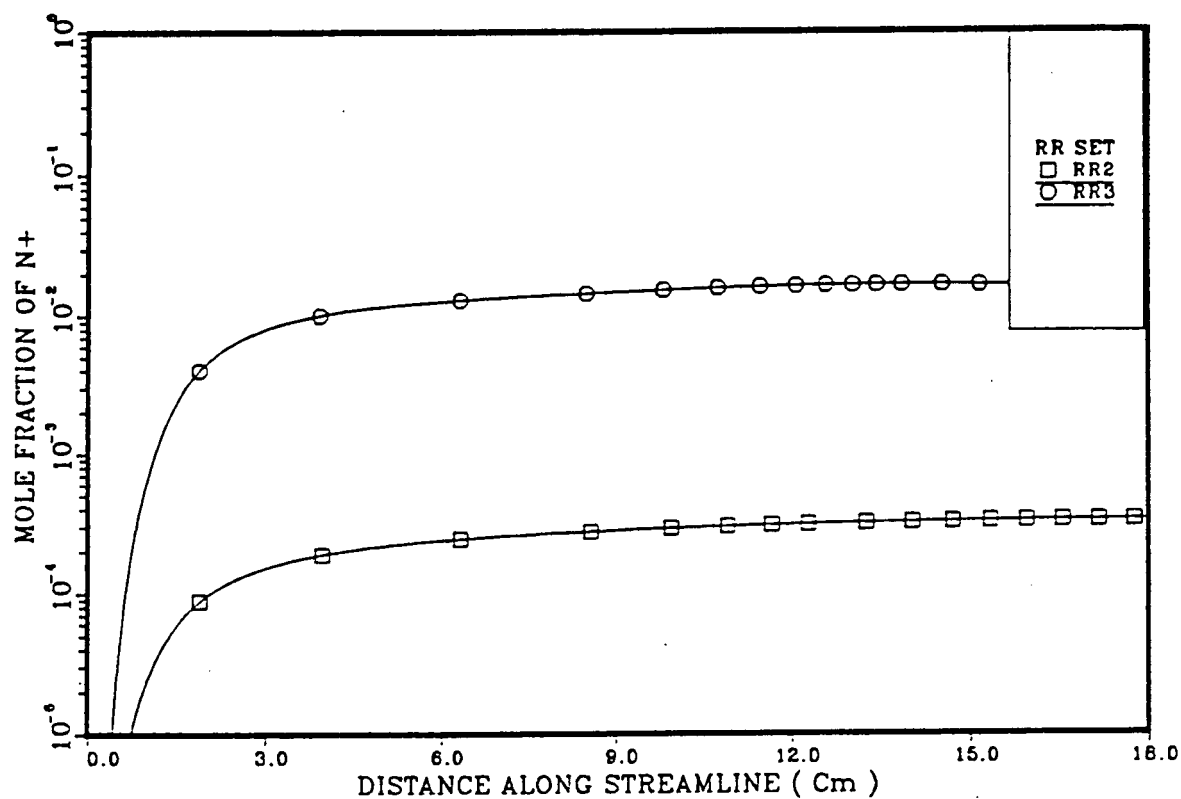


122(a)

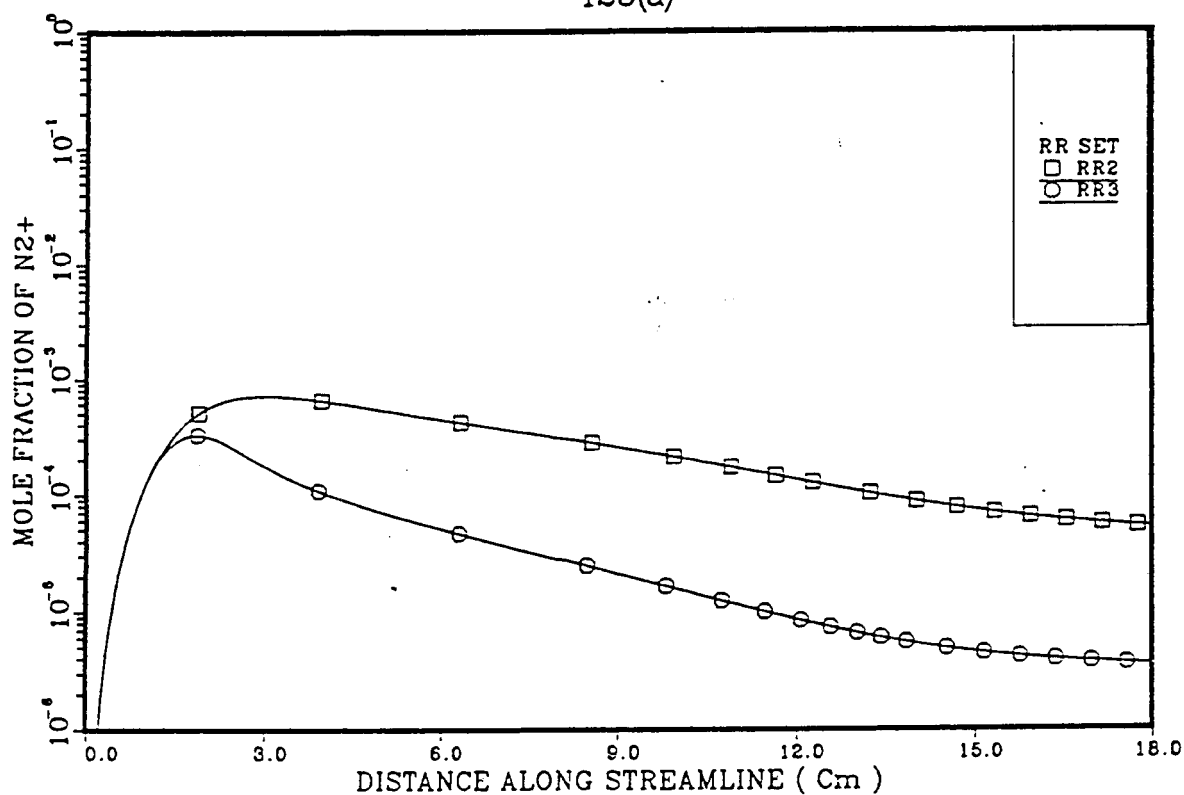


122(b)

FIGURES 122(a),122(b).PROFILES AT $V=7.7$ Km/s, CVDV



123(a)



123(b)

FIGURES 123(a),123(b).PROFILES AT $V=7.7$ Km/s, CVDV

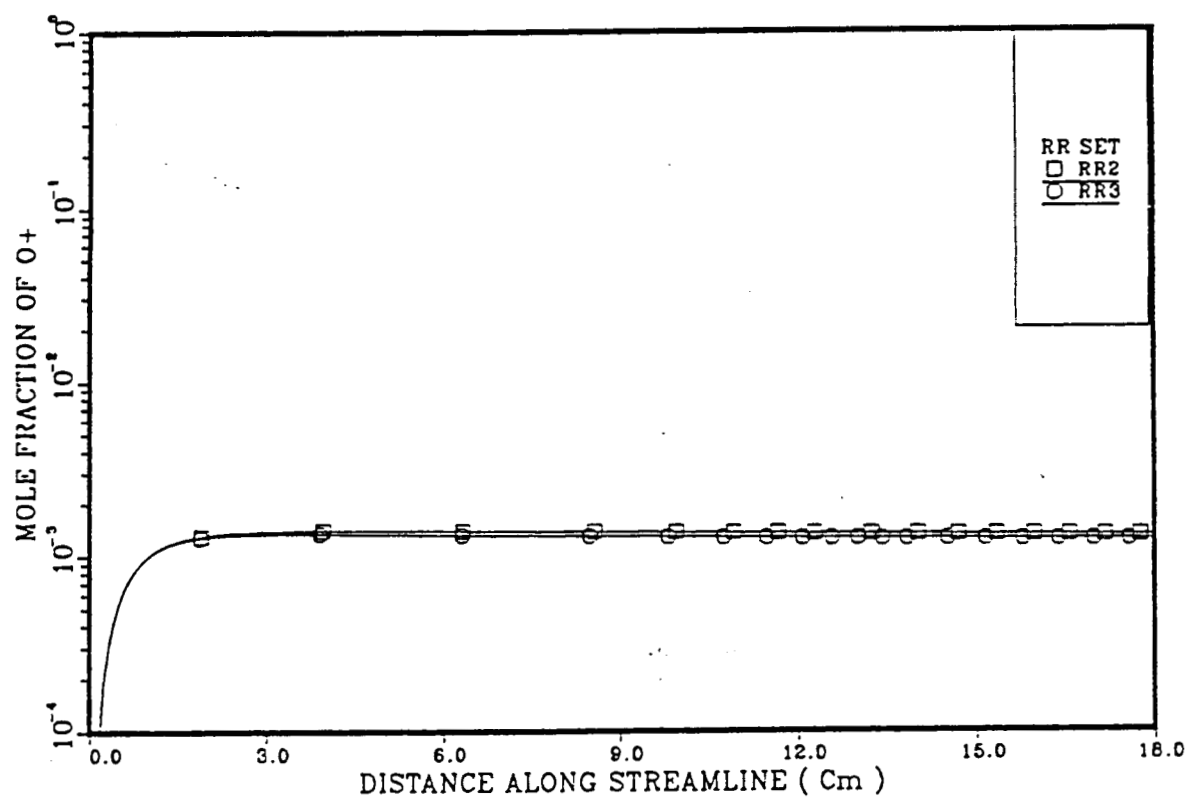
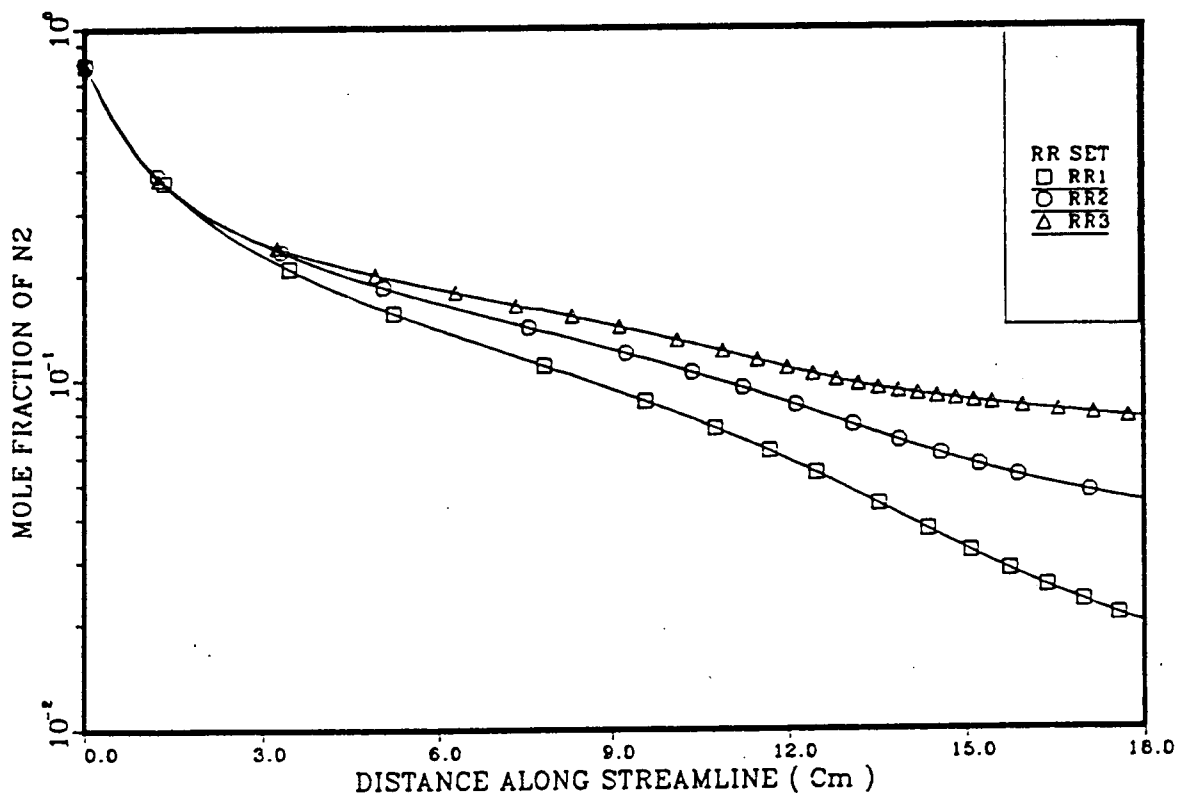
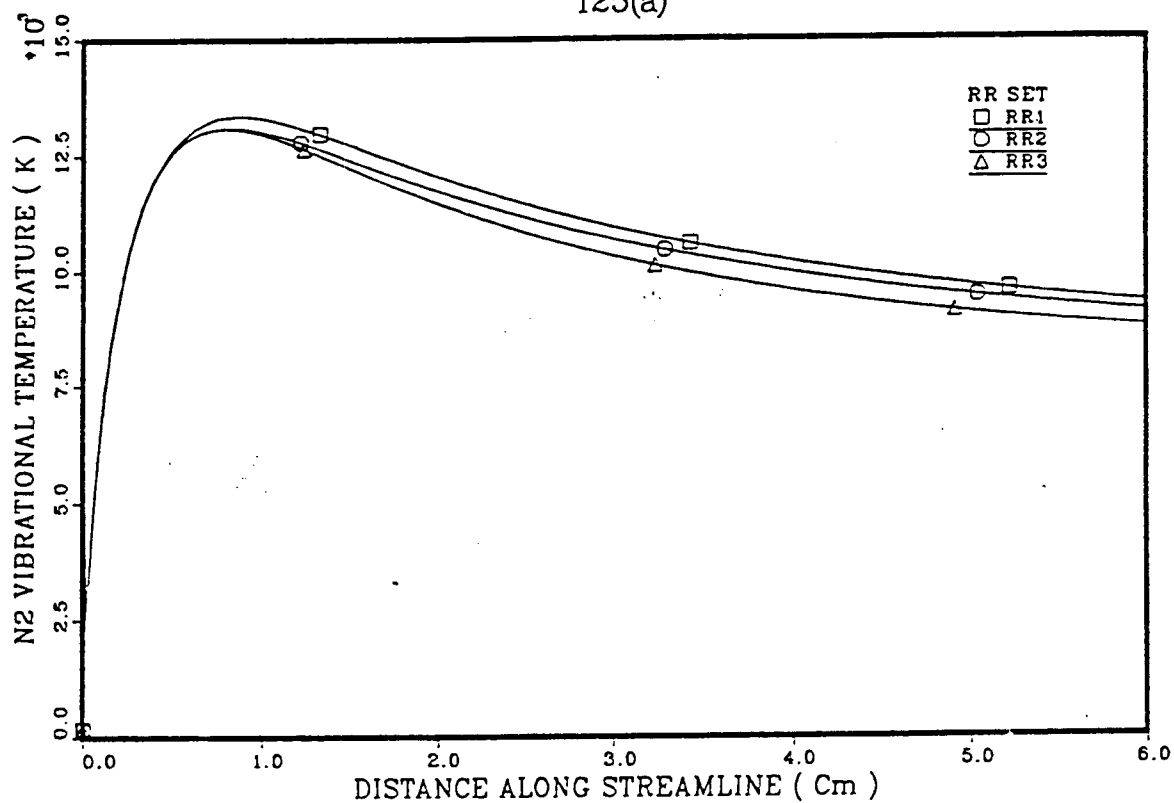


FIGURE 124.PROFILE AT V=7.7 Km/s, CVDV

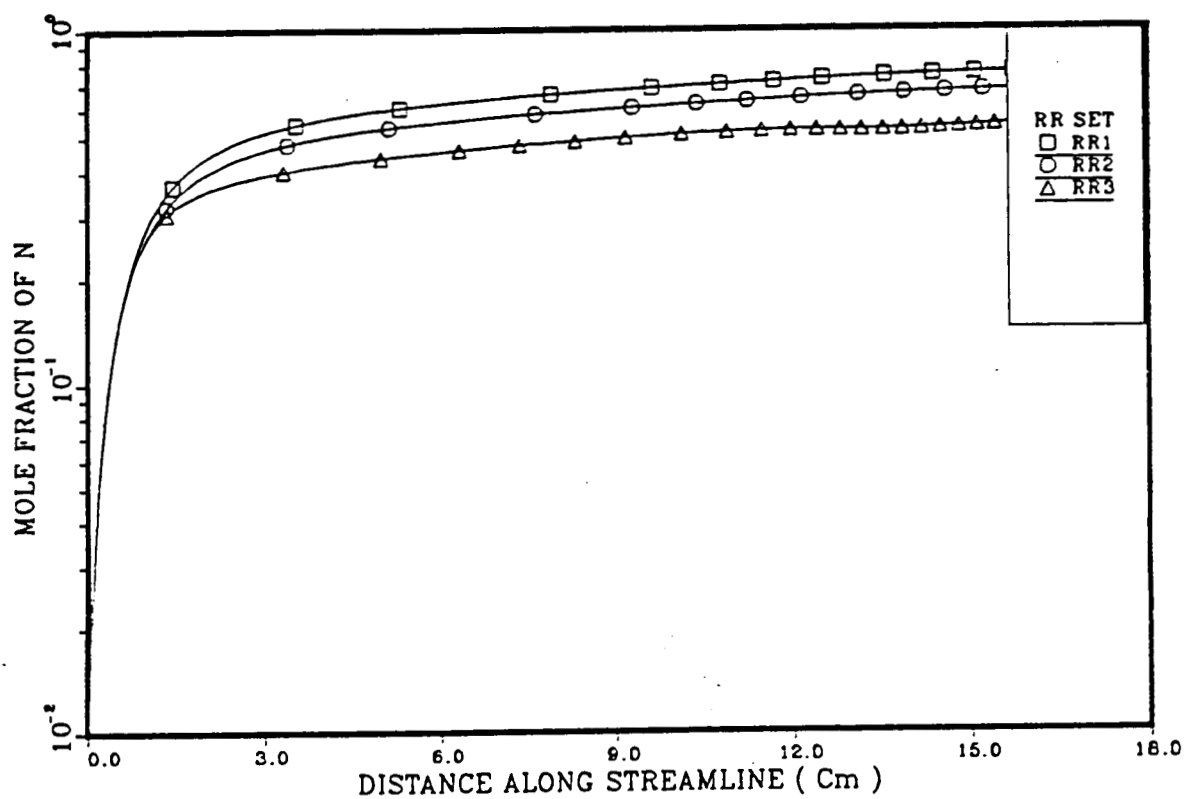


125(a)

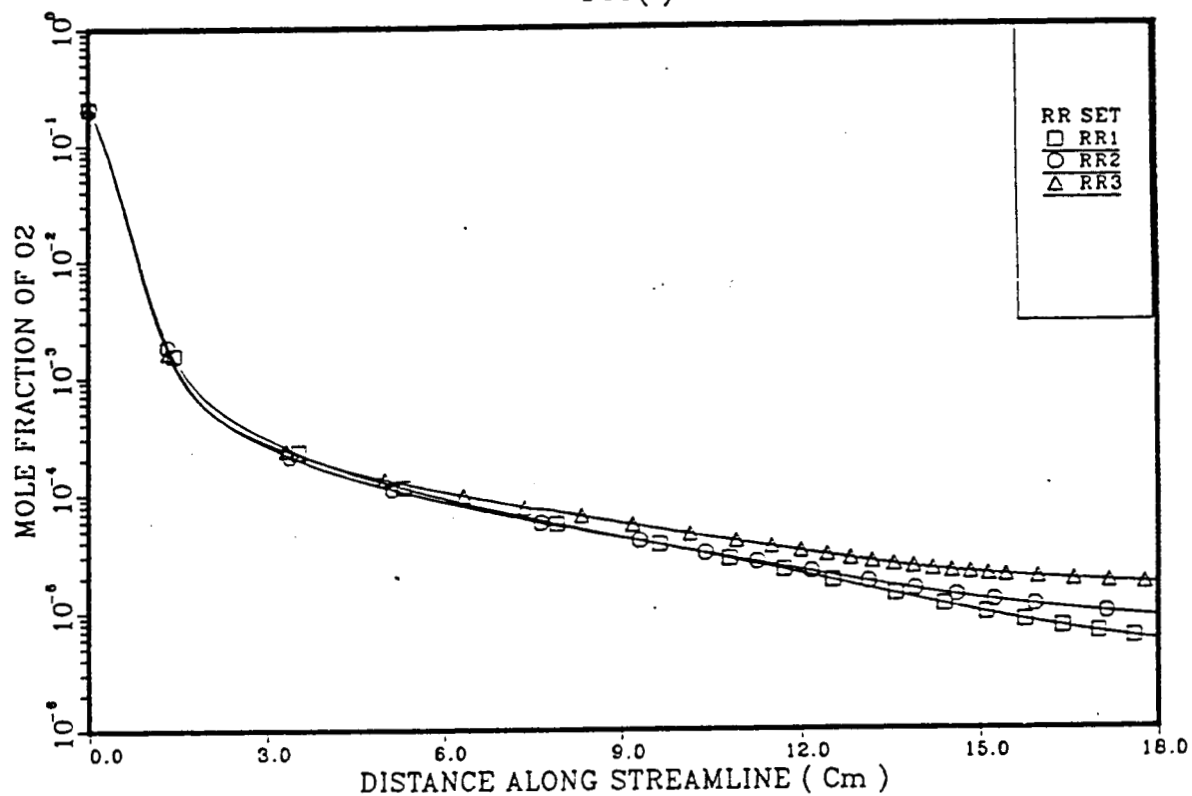


125(b)

FIGURES 125(a),125(b).PROFILES AT V=10 Km/s, PARK-L

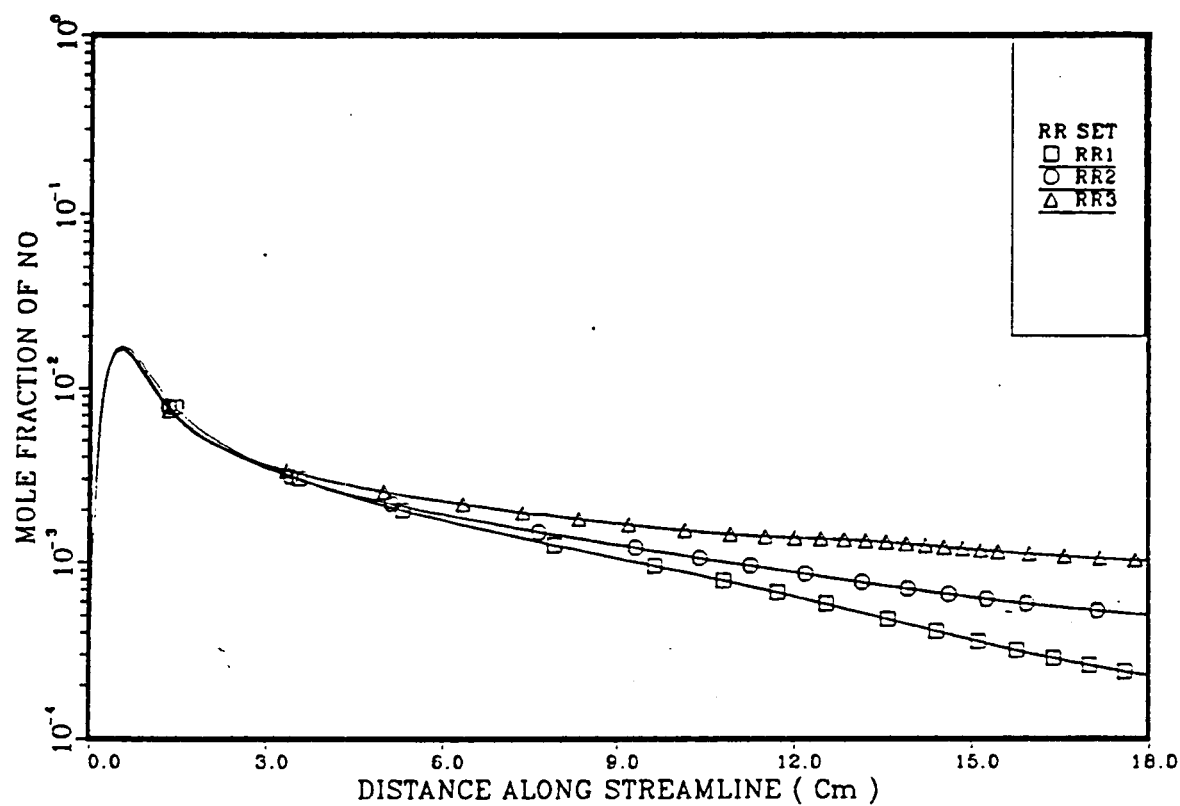


126(a)

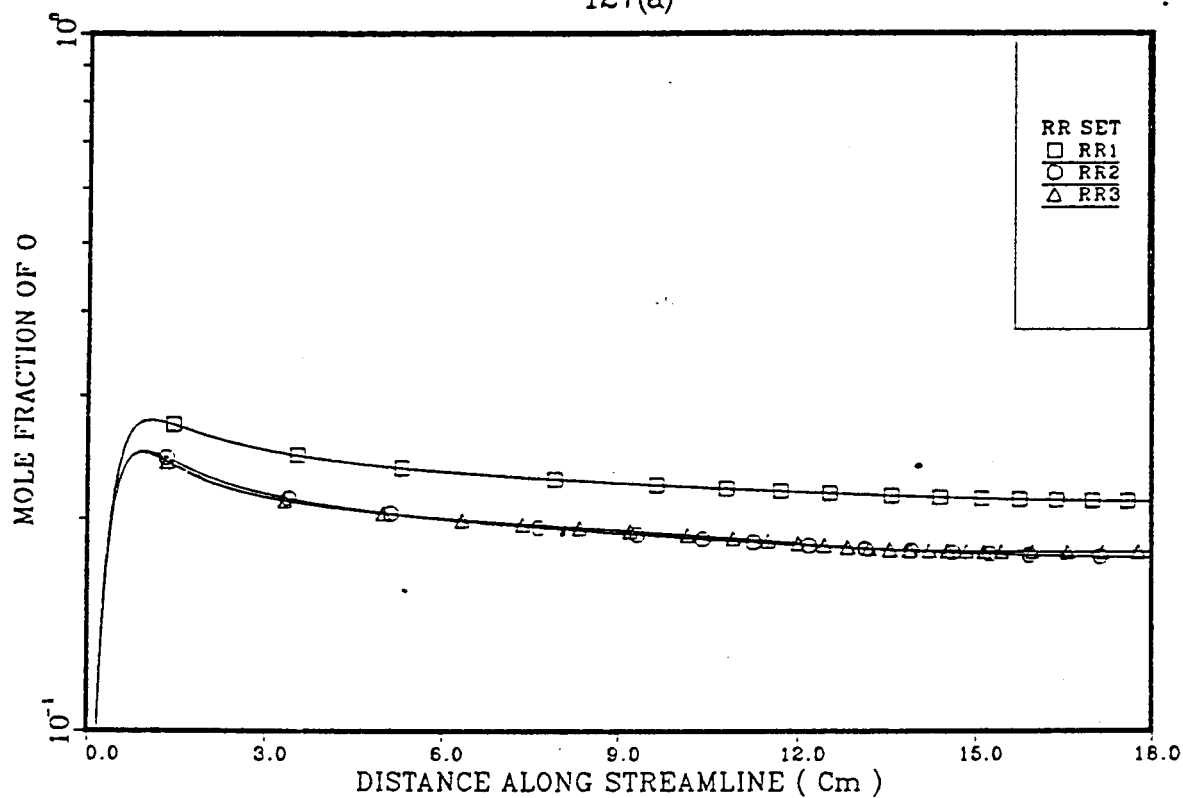


126(b)

FIGURES 126(a),126(b).PROFILES AT V=10 Km/s, PARK-L

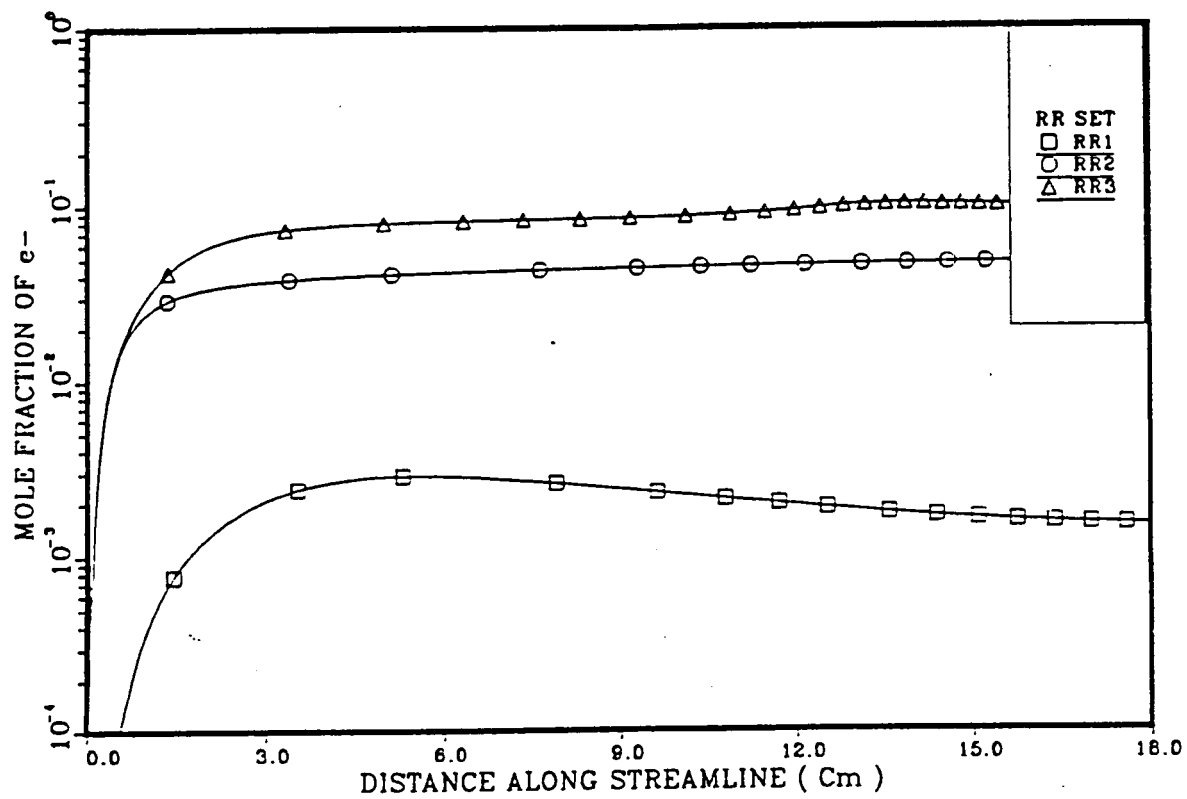


127(a)

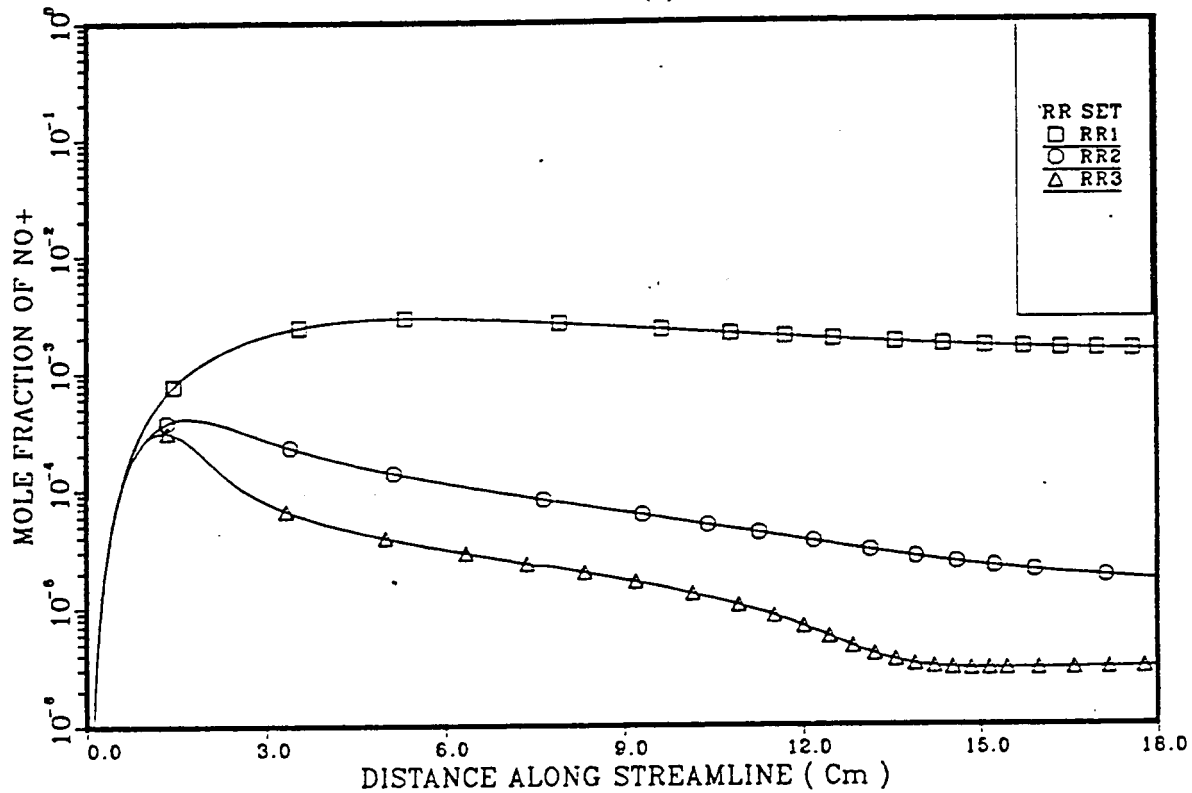


127(b)

FIGURES 127(a),127(b).PROFILES AT $V=10$ Km/s, PARK-L



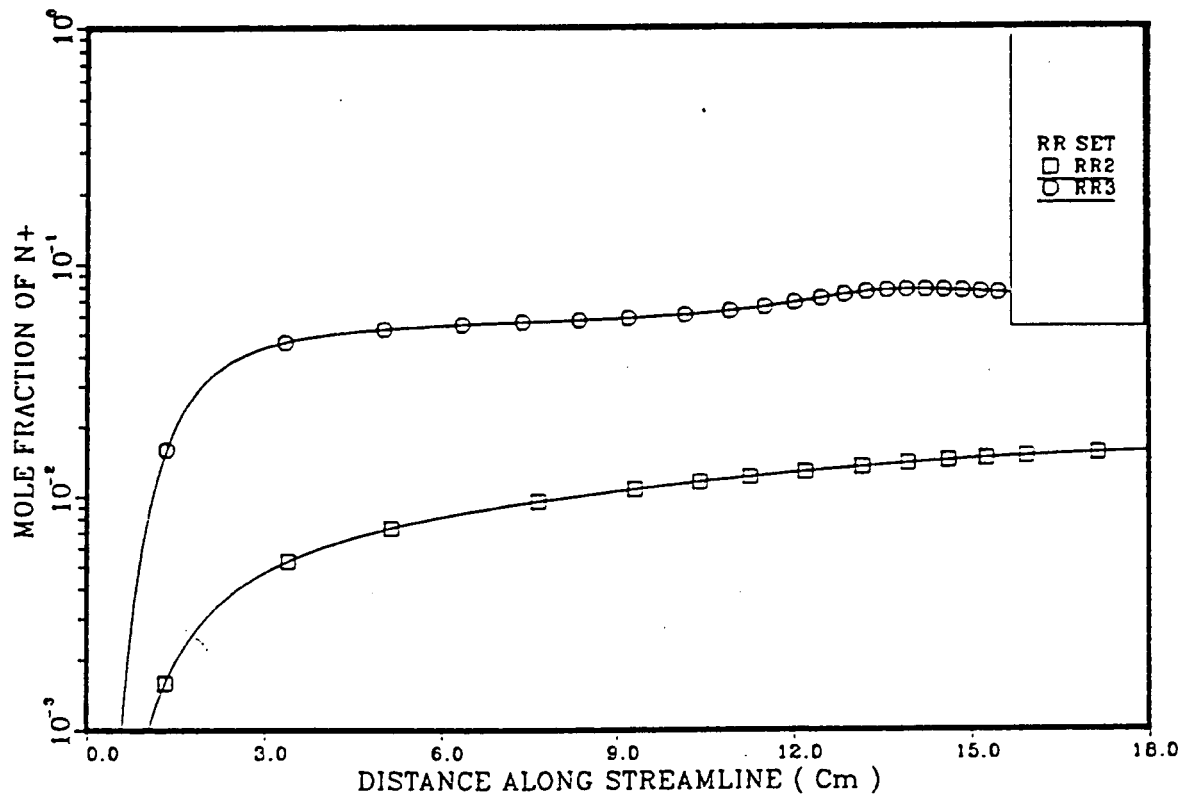
128(a)



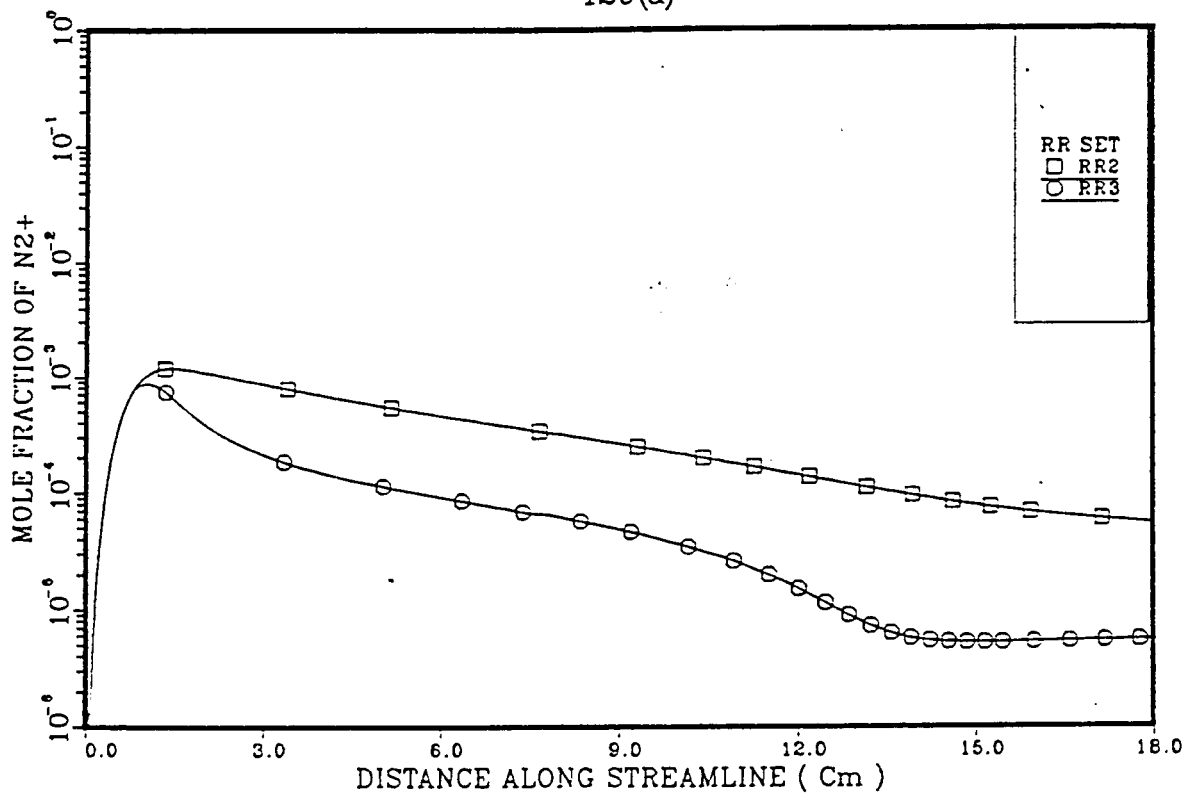
128(b)

FIGURES 128(a),128(b).PROFILES AT $V=10$ Km/s, PARK-L

ORIGINAL PAGE IS
OF POOR QUALITY



129(a)



129(b)

FIGURES 129(a),129(b).PROFILES AT $V=10$ Km/s, PARK-L

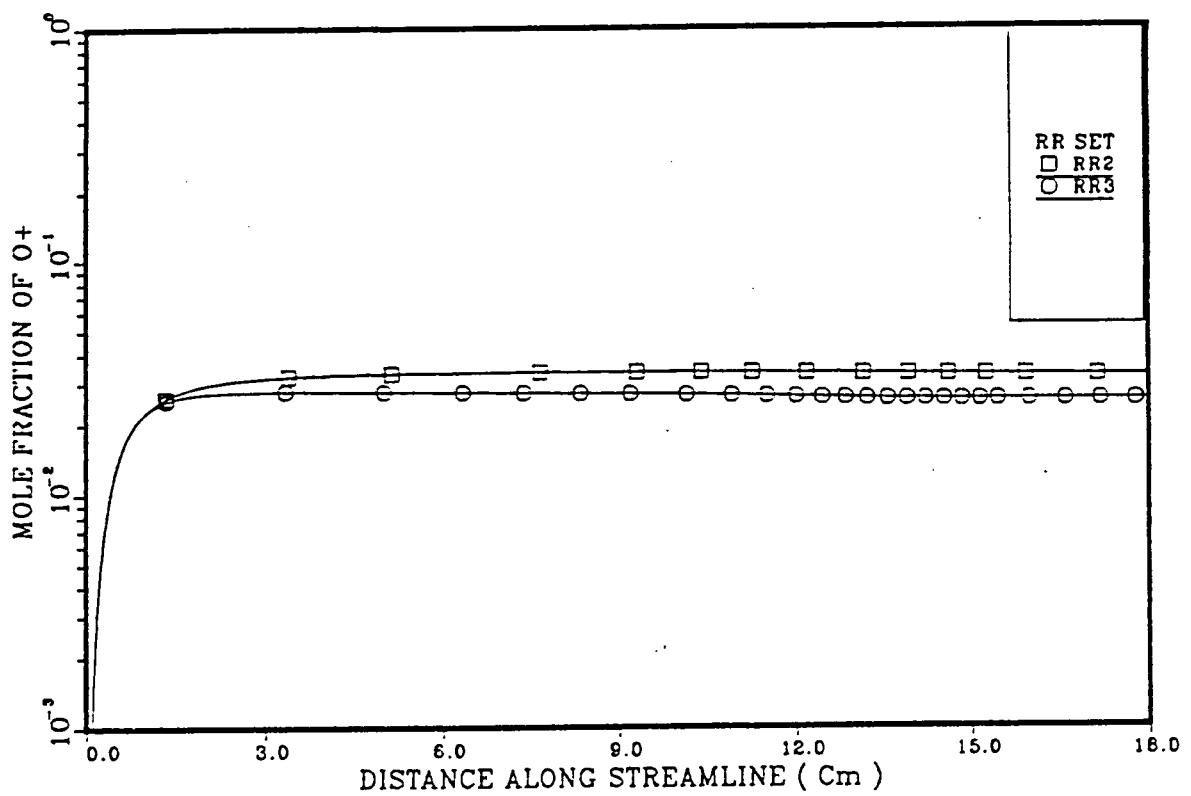
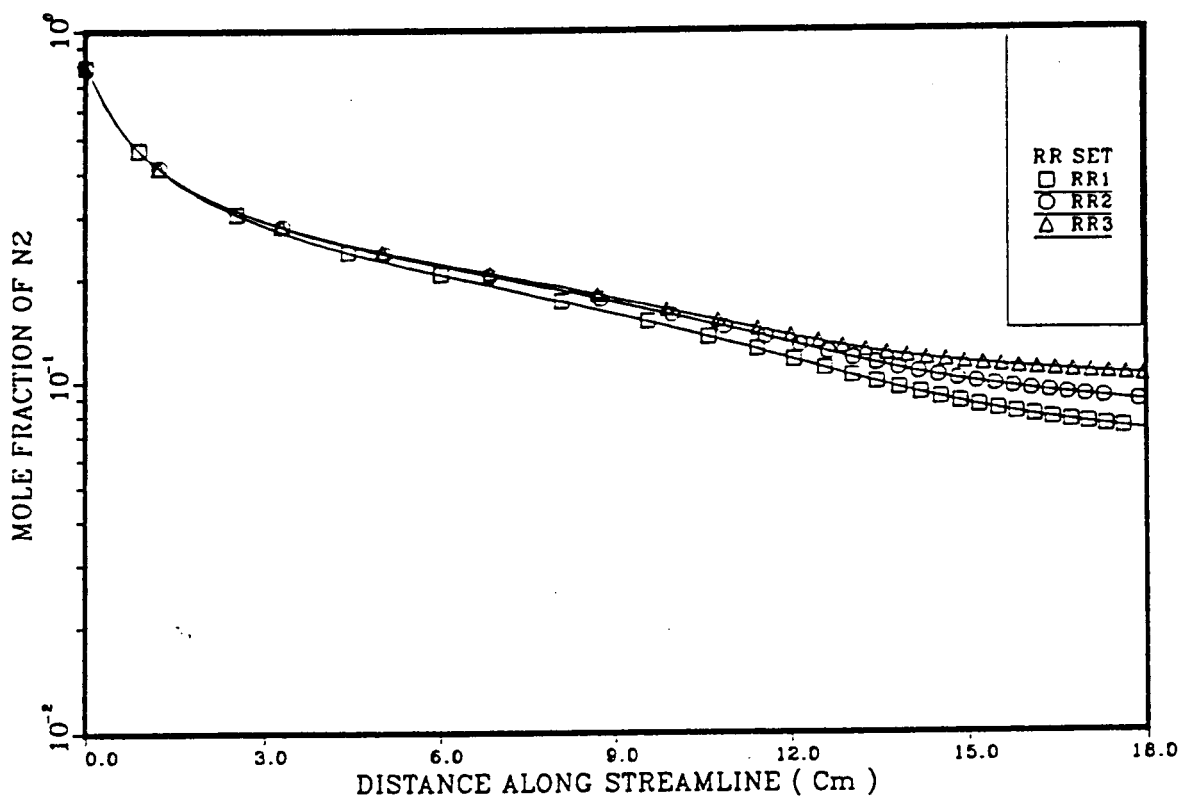
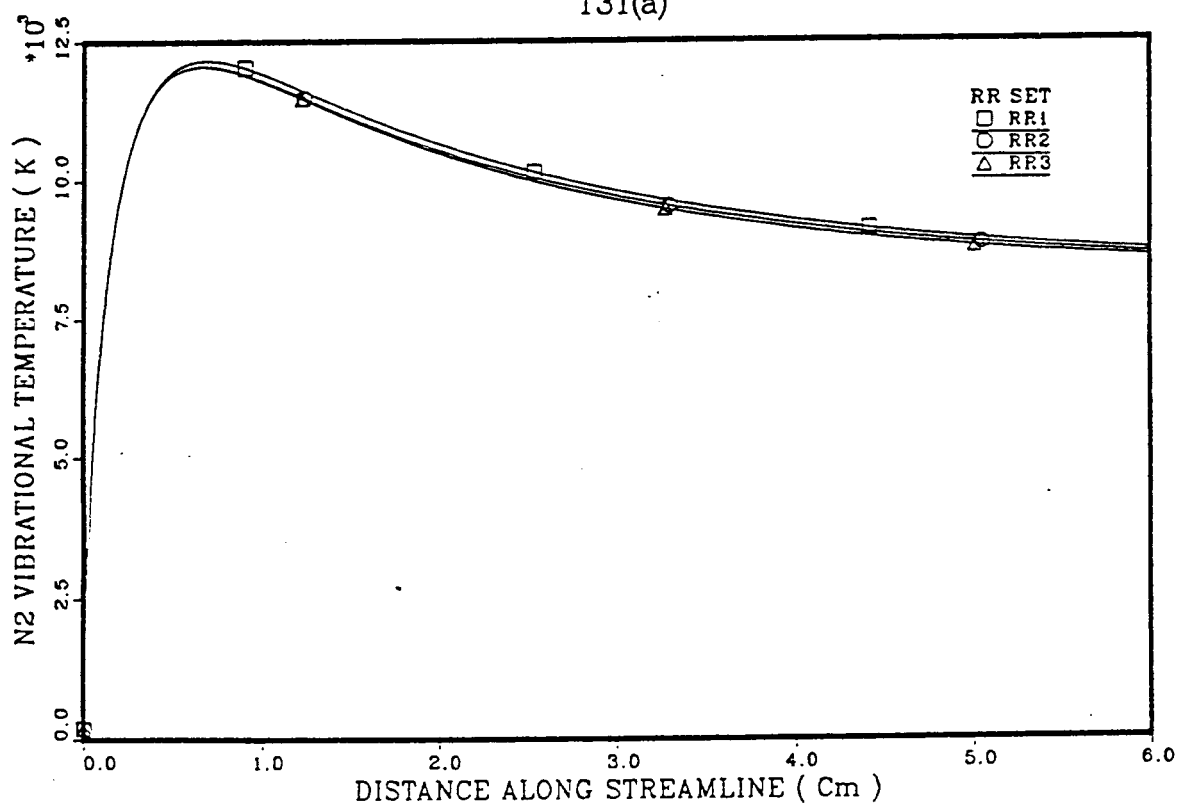


FIGURE 130.PROFILE AT V=10 Km/s, PARK-L

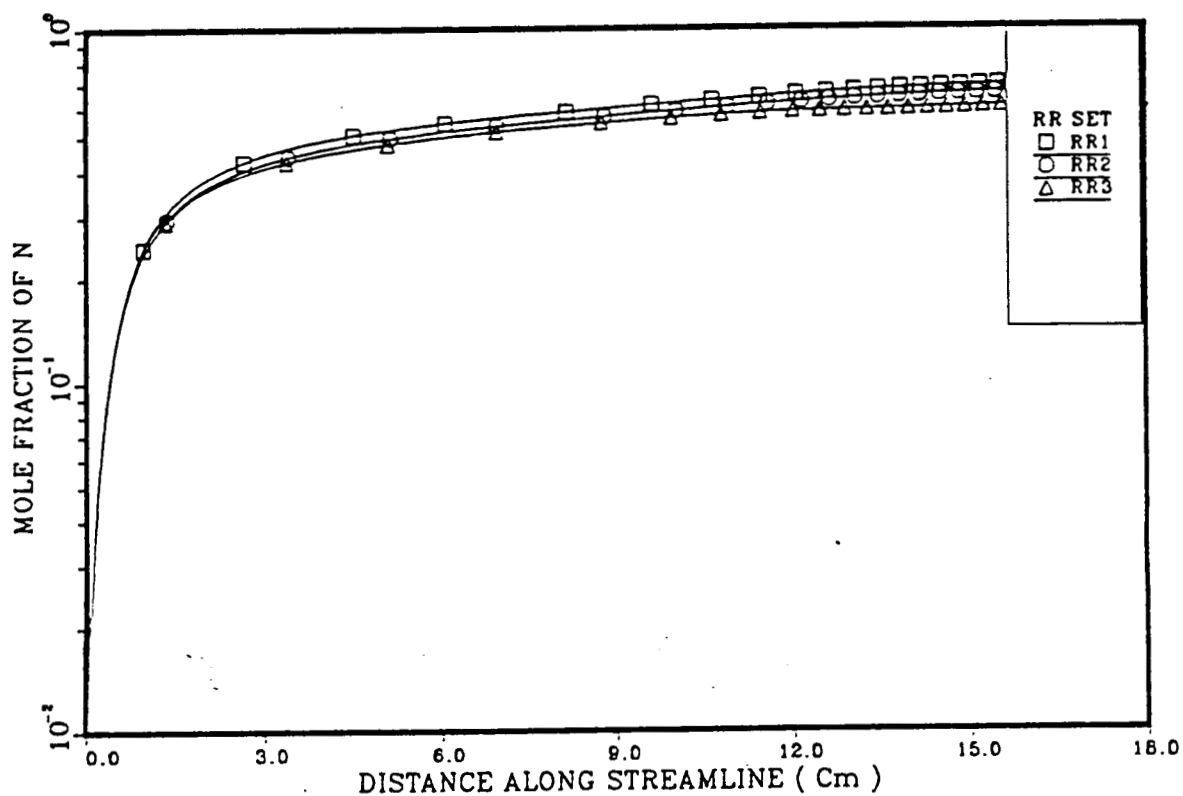


131(a)

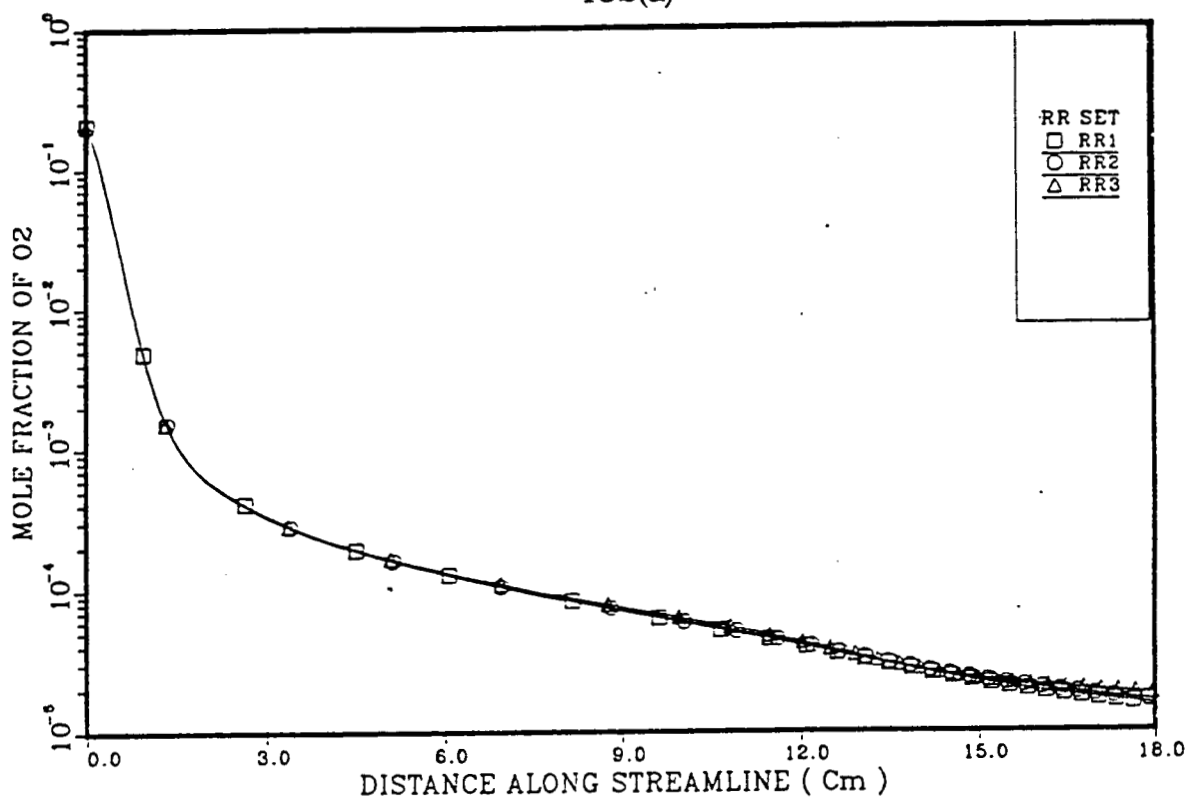


131(b)

FIGURES 131(a),131(b).PROFILES AT V=8.9 Km/s. PARK-L

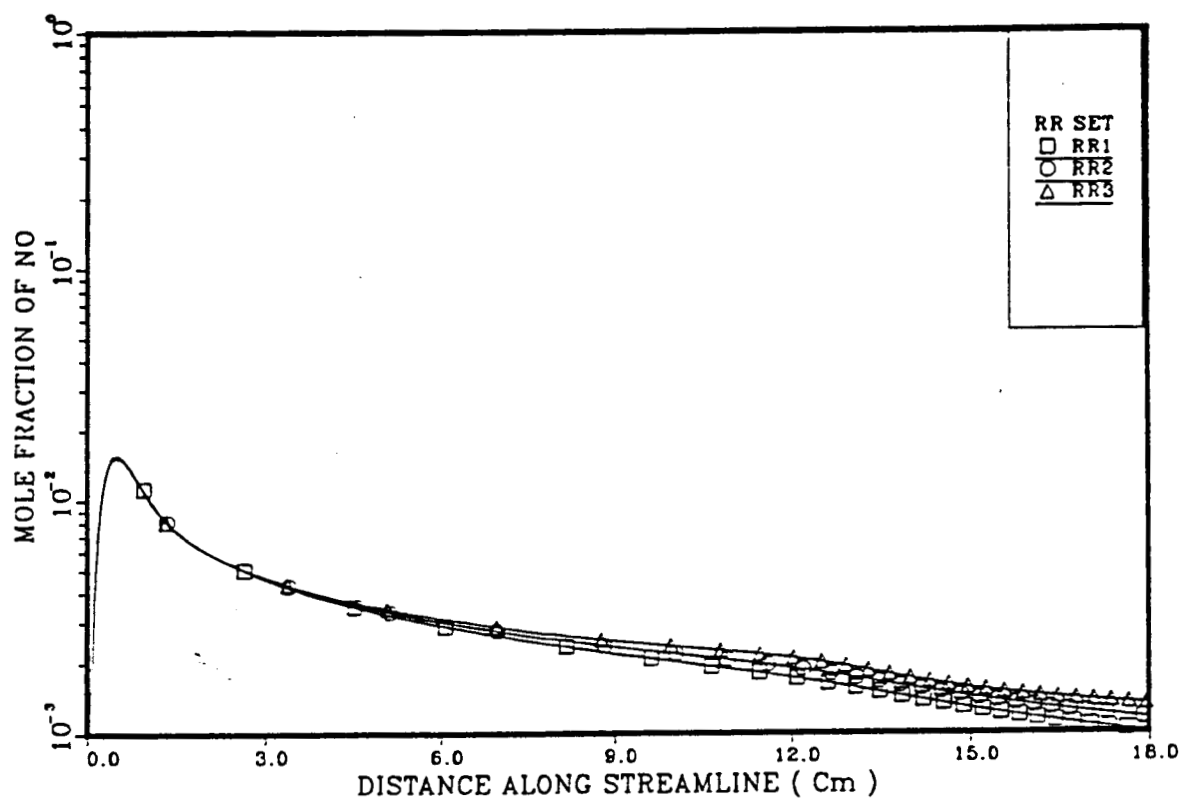


132(a)

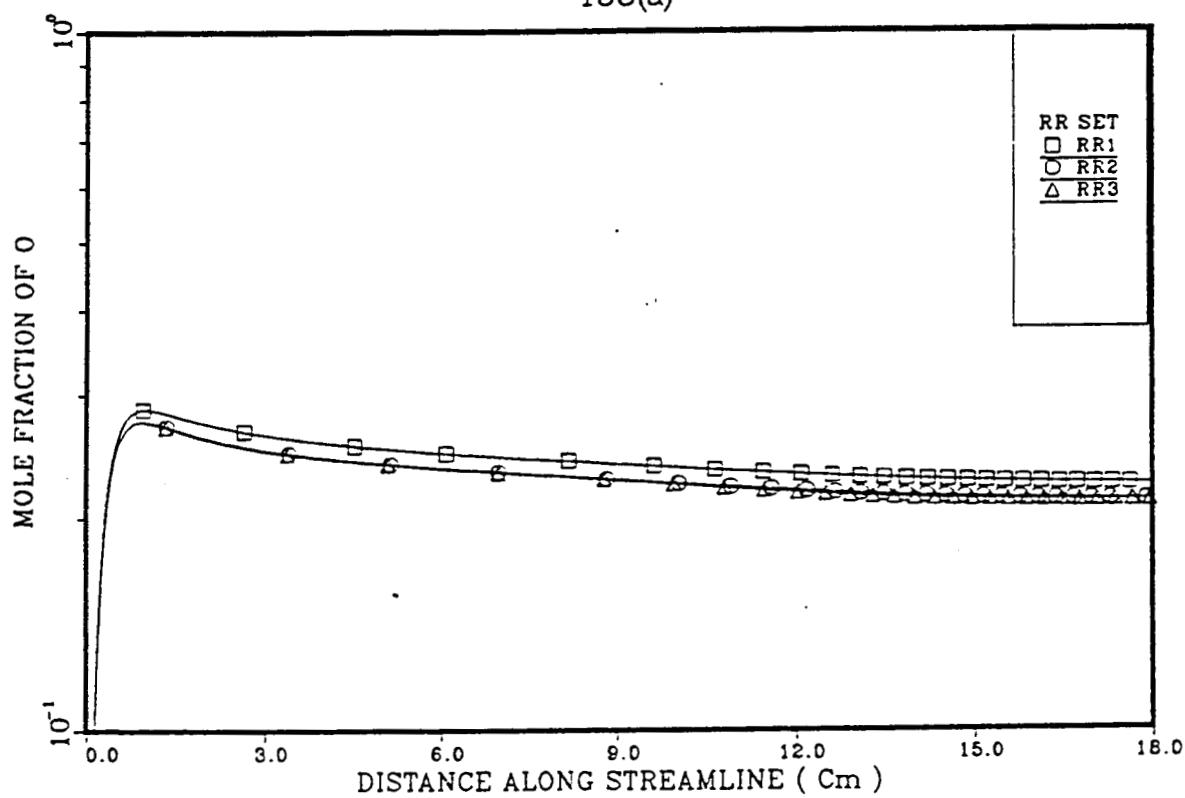


132(b)

FIGURES 132(a),132(b).PROFILES AT V=8.9 Km/s, PARK-L

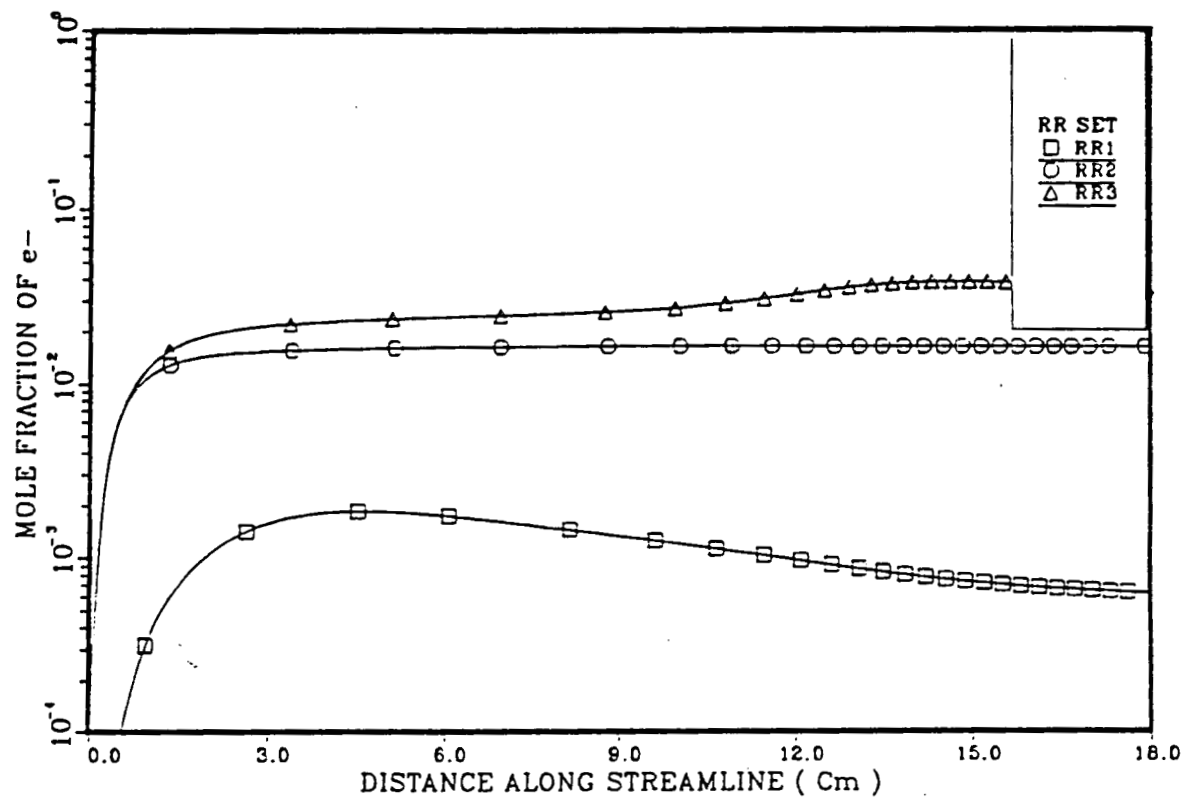


133(a)

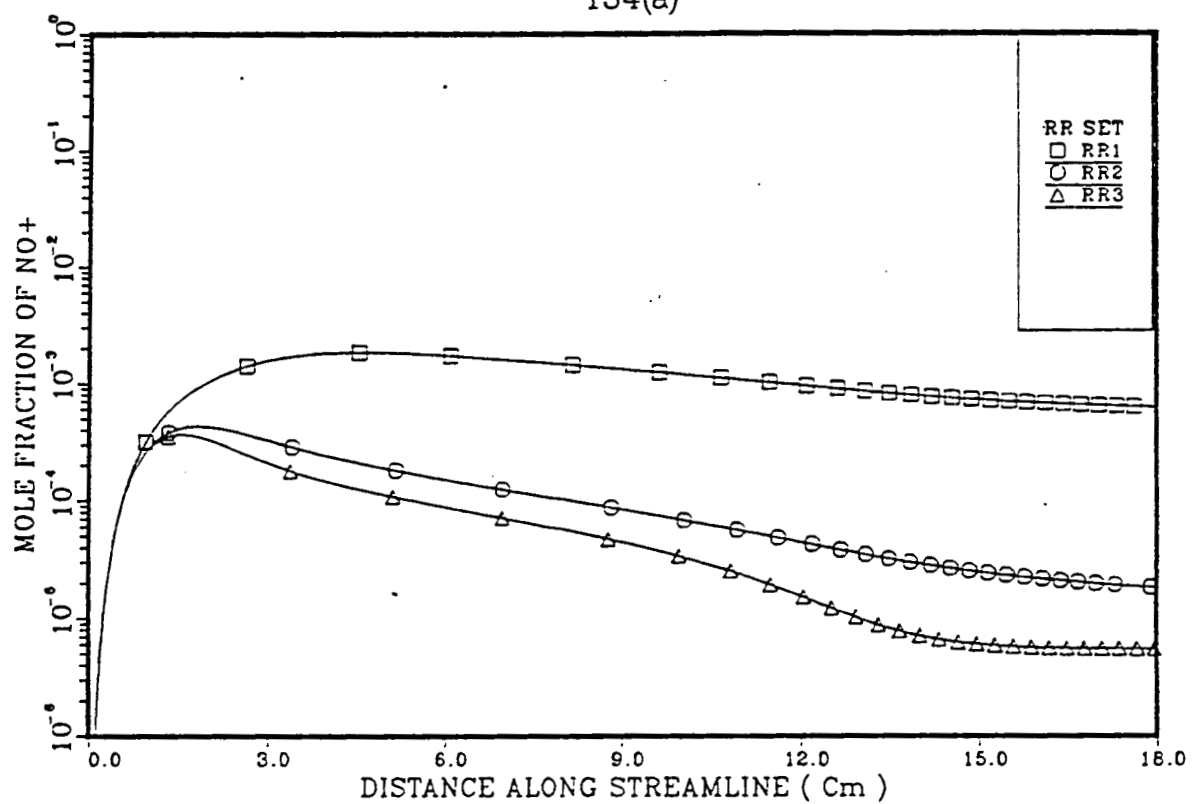


133(b)

FIGURES 133(a),133(b).PROFILES AT V=8.9 Km/s, PARK-L

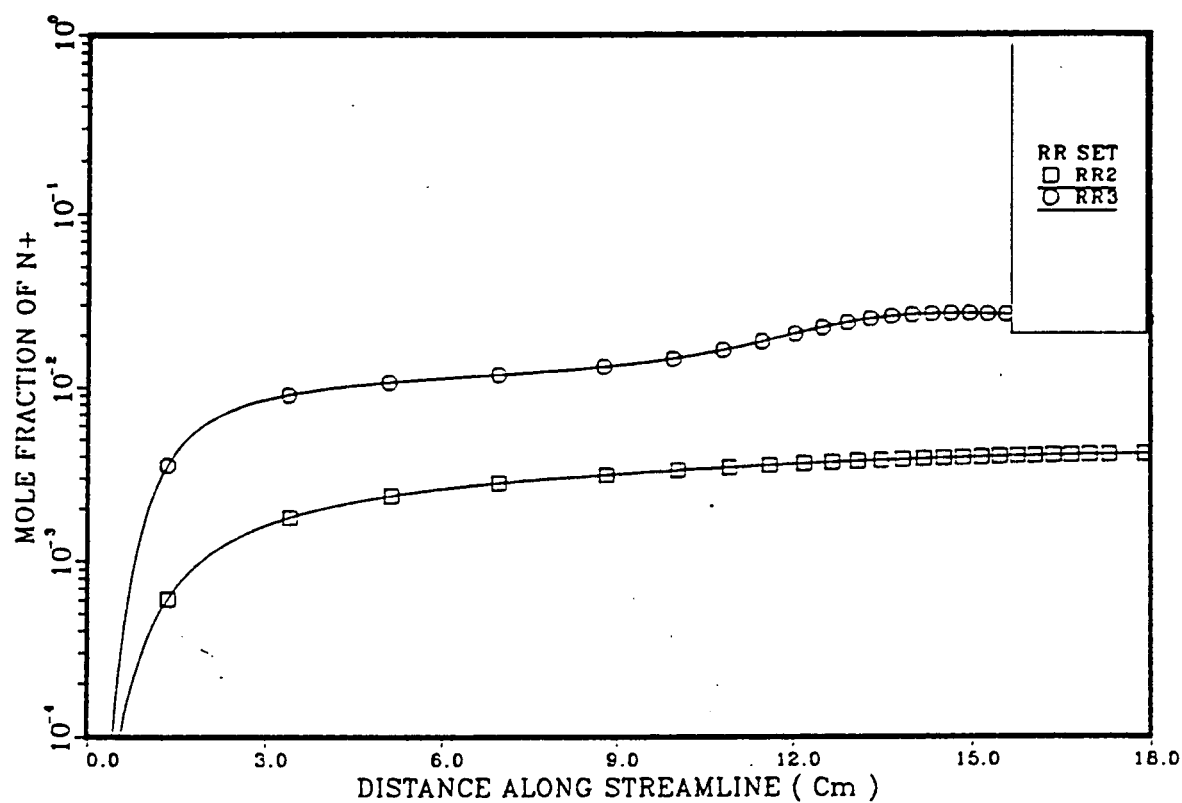


134(a)

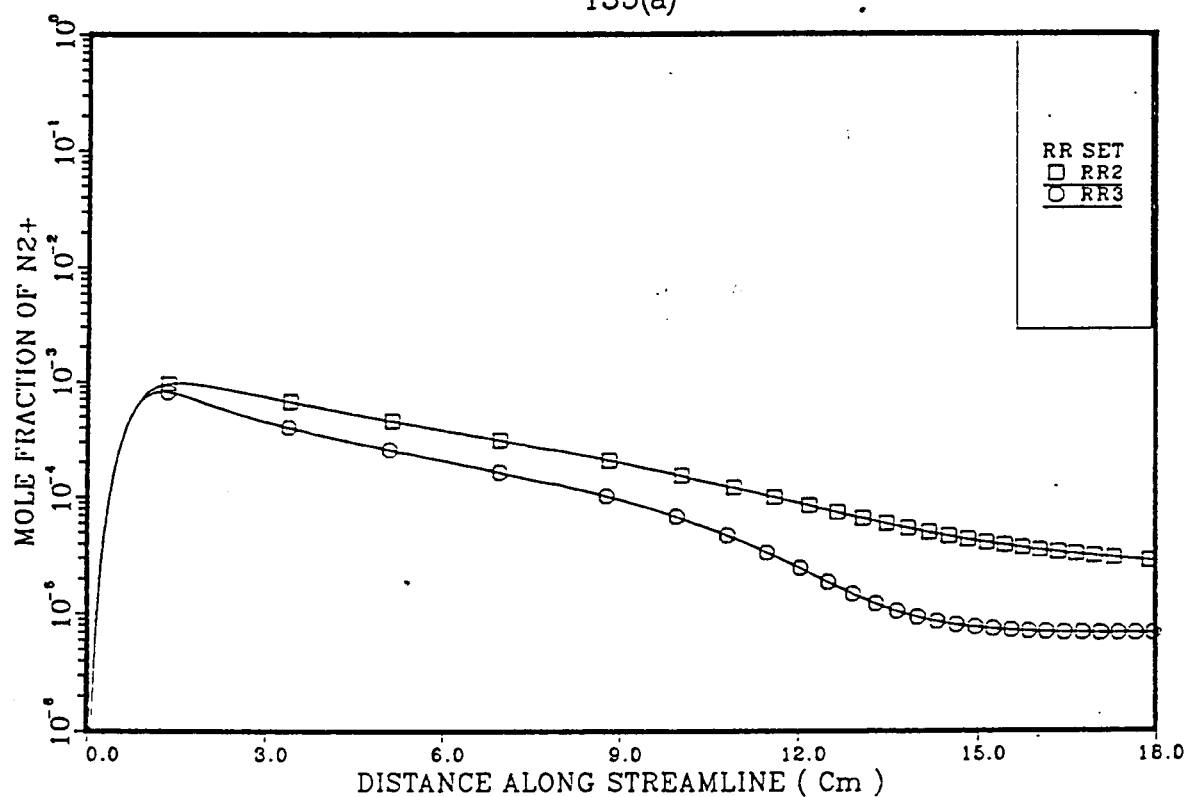


134(b)

FIGURES 134(a),134(b).PROFILES AT $V=8.9$ Km/s, PARK-L



135(a)



135(b)

FIGURES 135(a),135(b).PROFILES AT $V=8.9$ Km/s, PARK-L

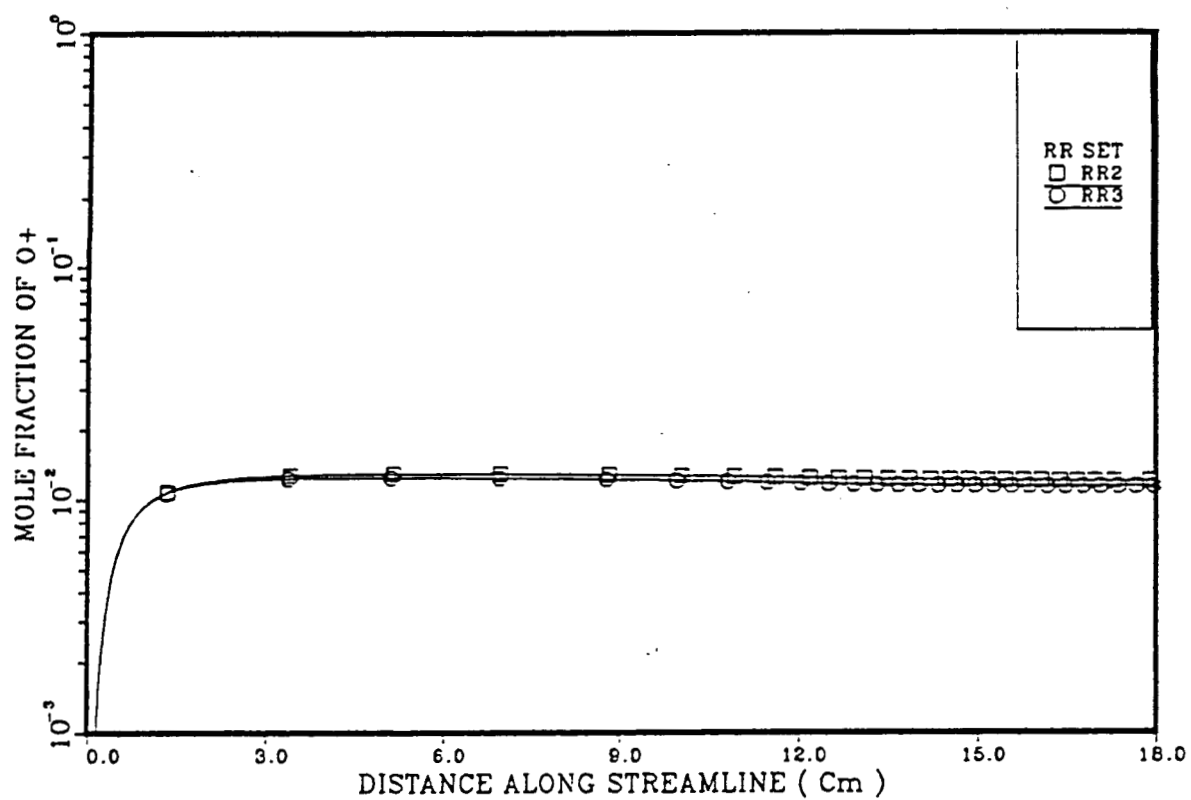
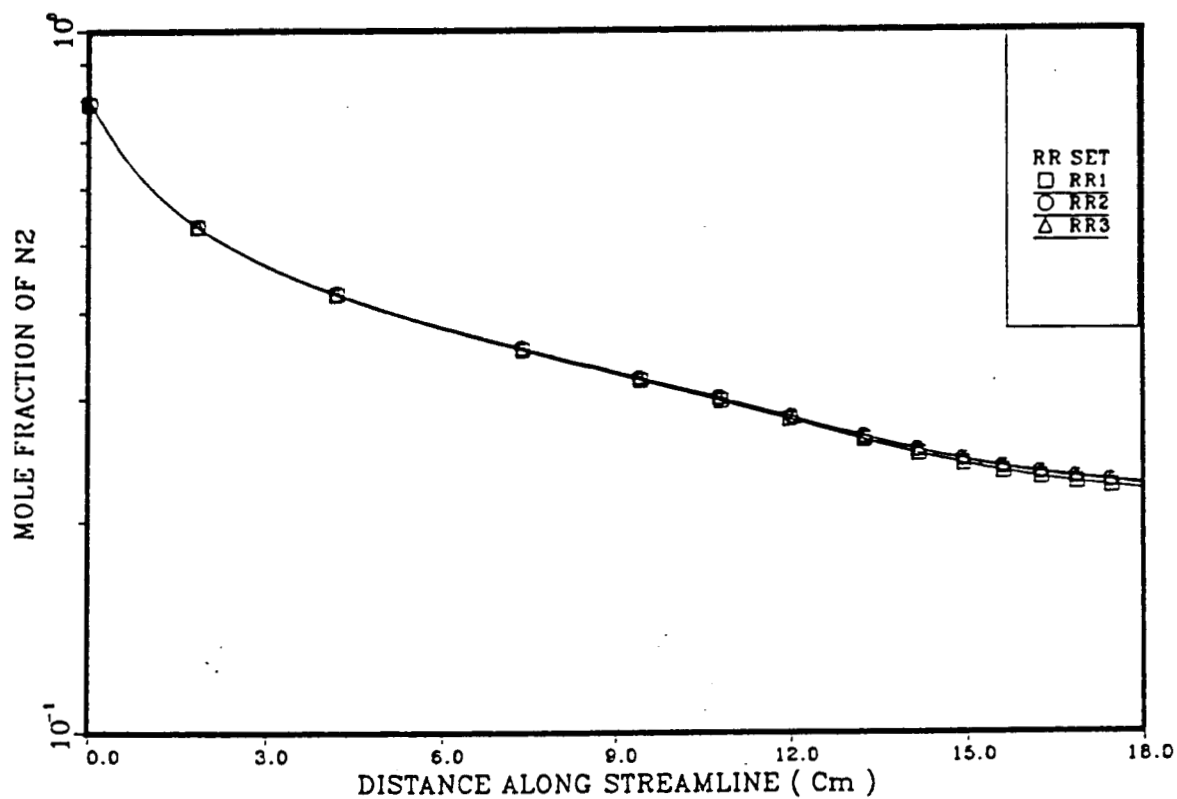
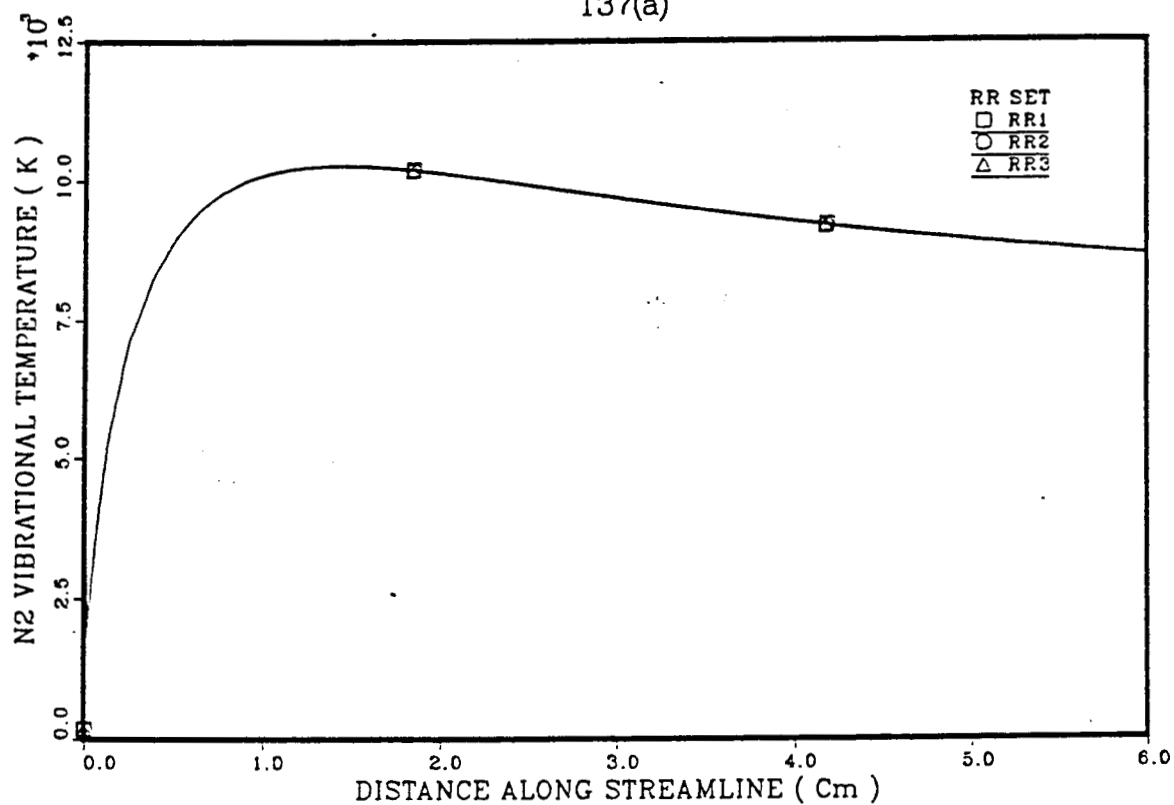


FIGURE 136.PROFILE AT V=8.9 Km/s, PARK-L

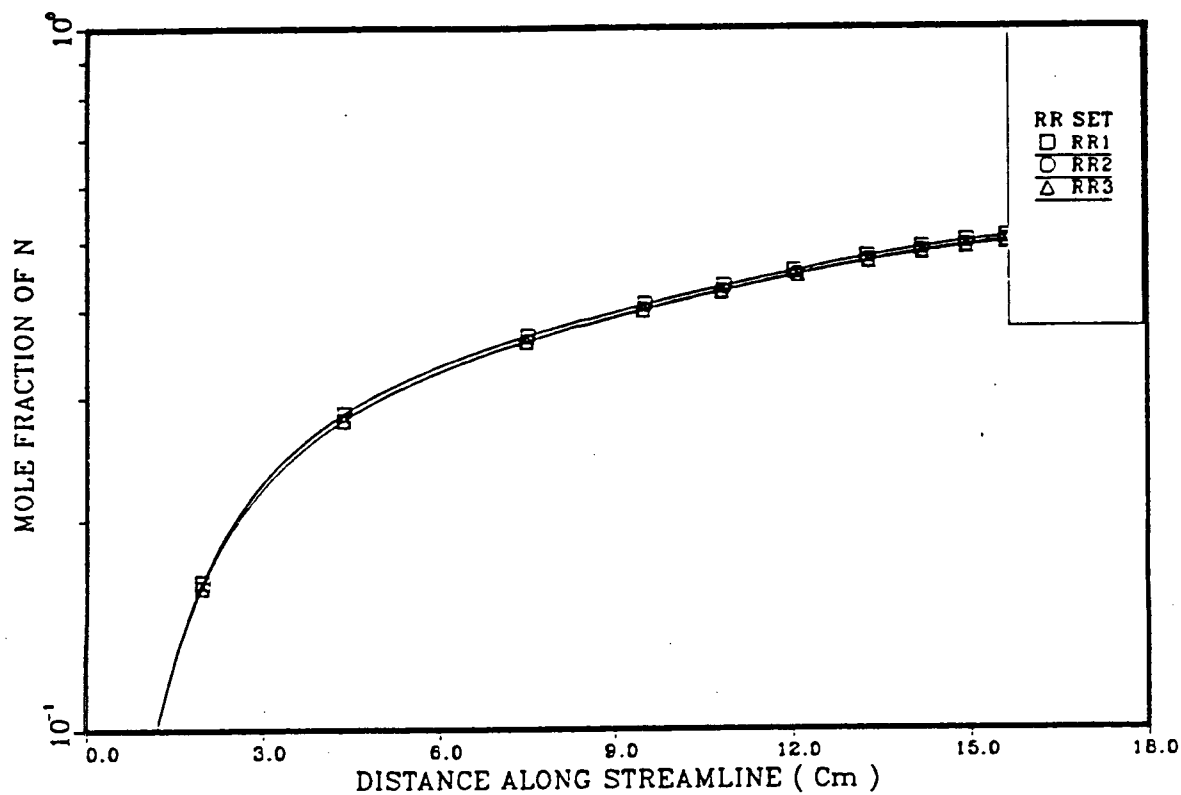


137(a)

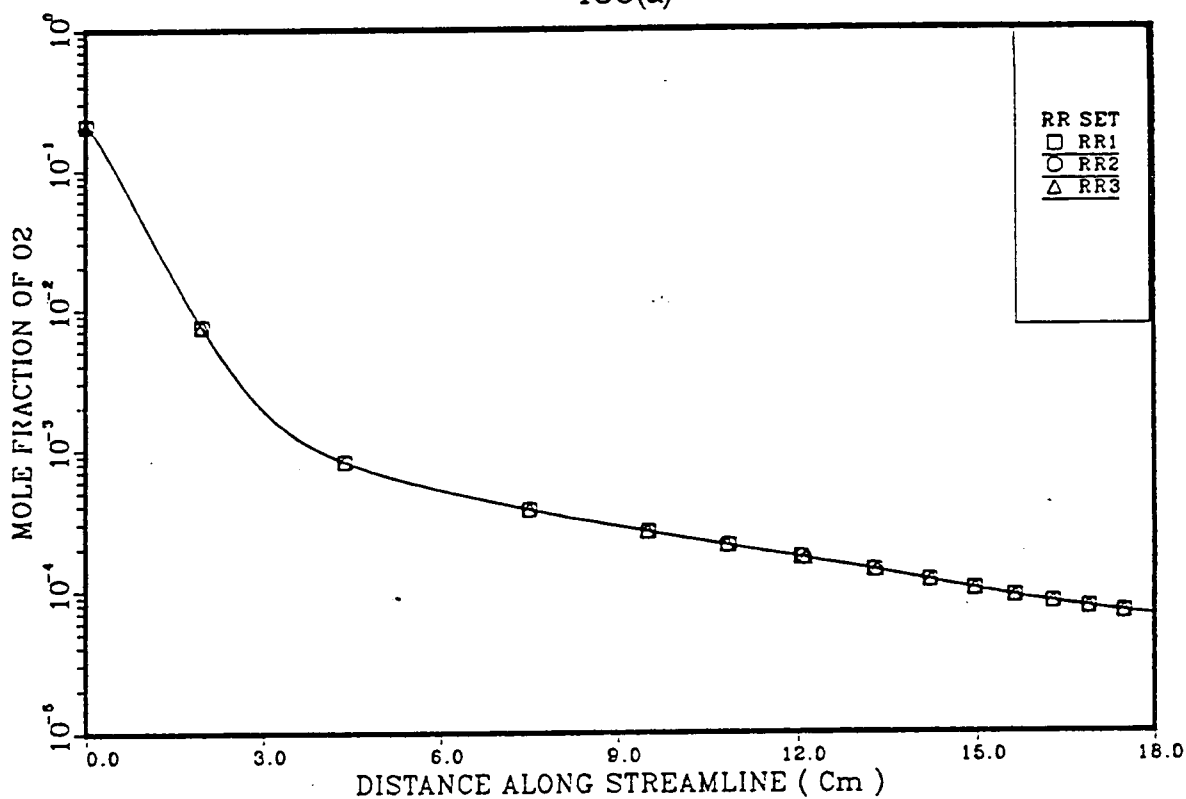


137(b)

FIGURES 137(a),137(b).PROFILES AT V=7.7 Km/s, PARK-L

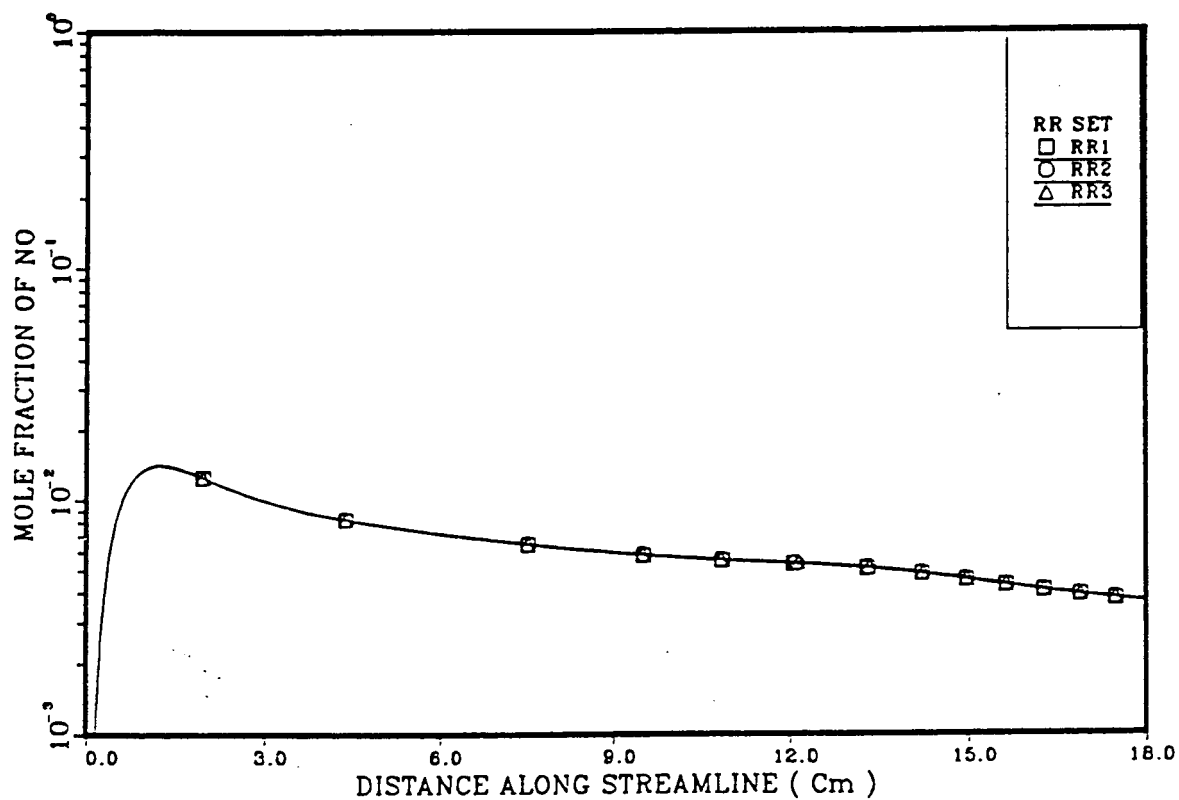


138(a)

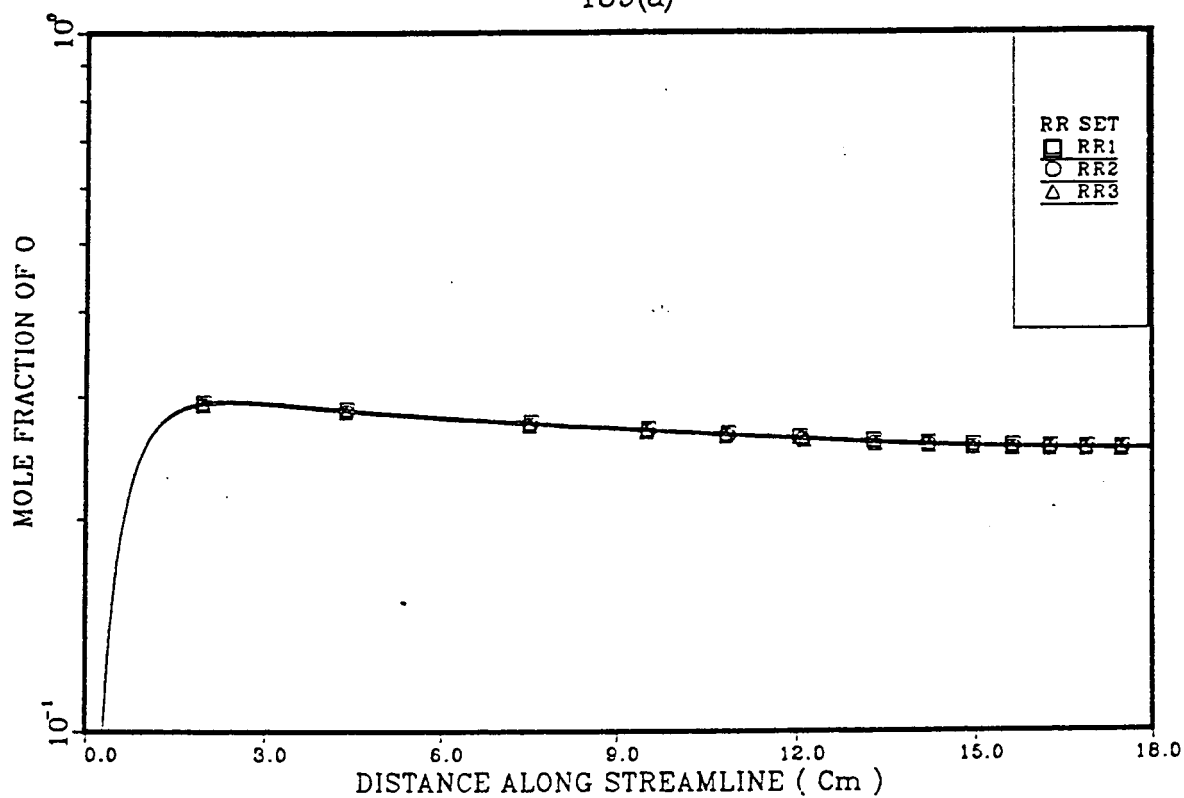


138(b)

FIGURES 138(a),138(b).PROFILES AT V=7.7 Km/s, PARK-L

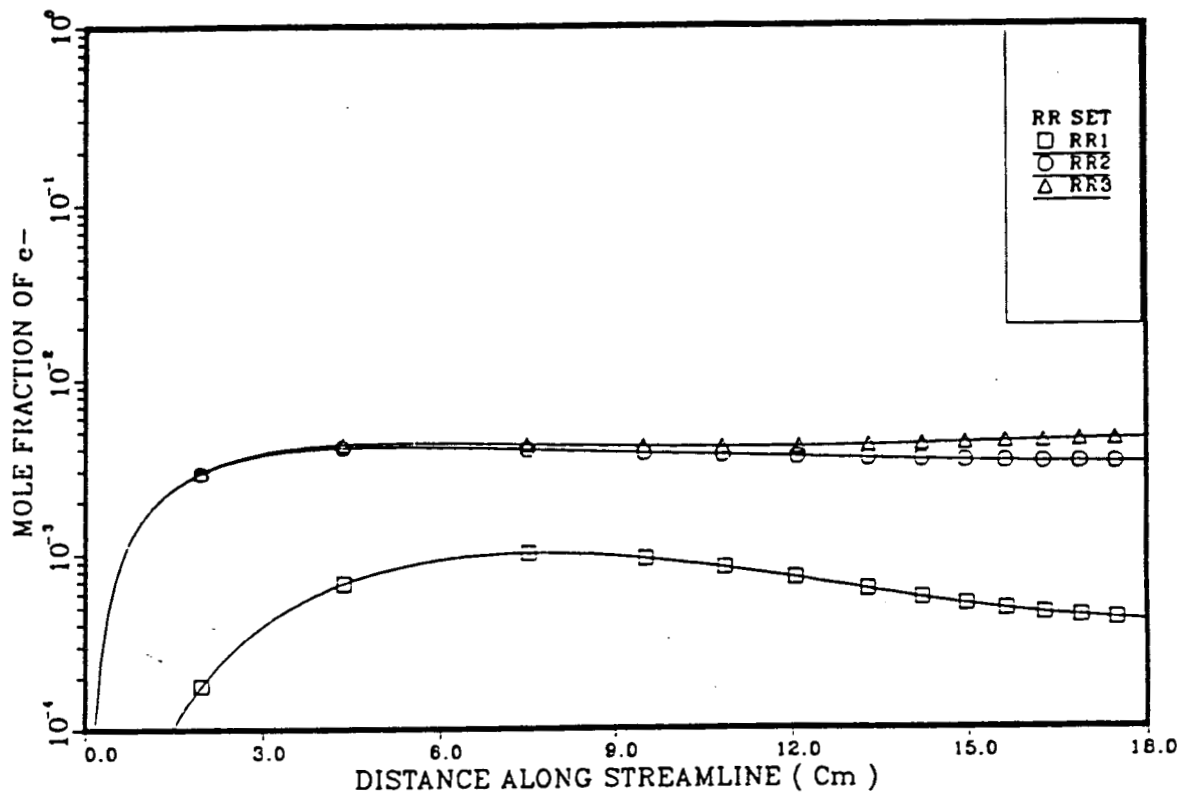


139(a)

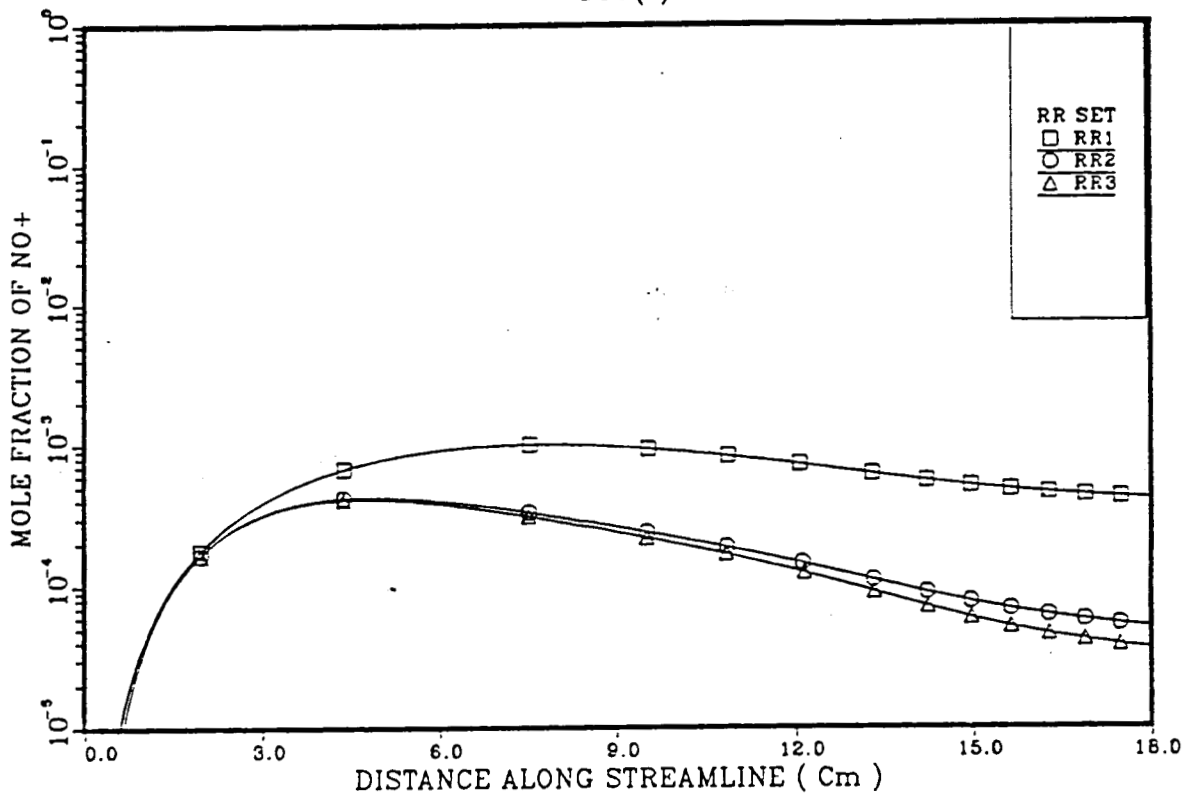


139(b)

FIGURES 139(a),139(b).PROFILES AT V=7.7 Km/s, PARK-L

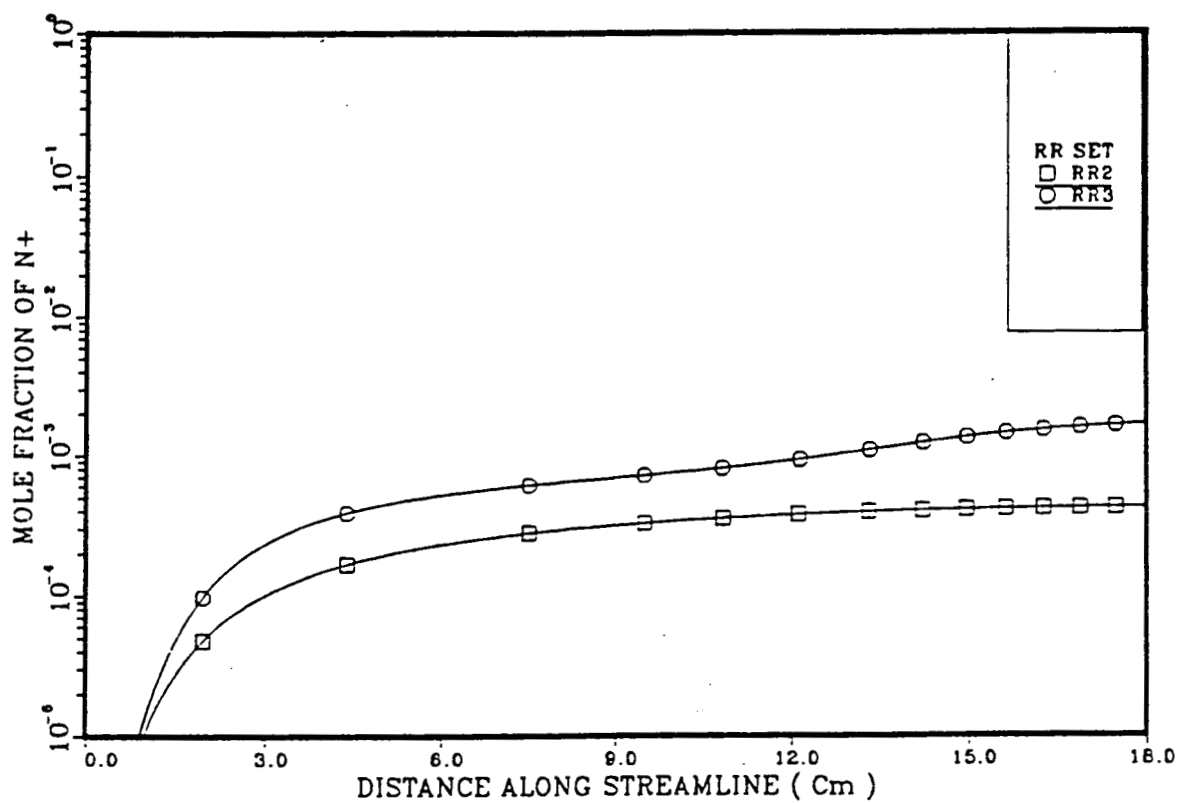


140(a)

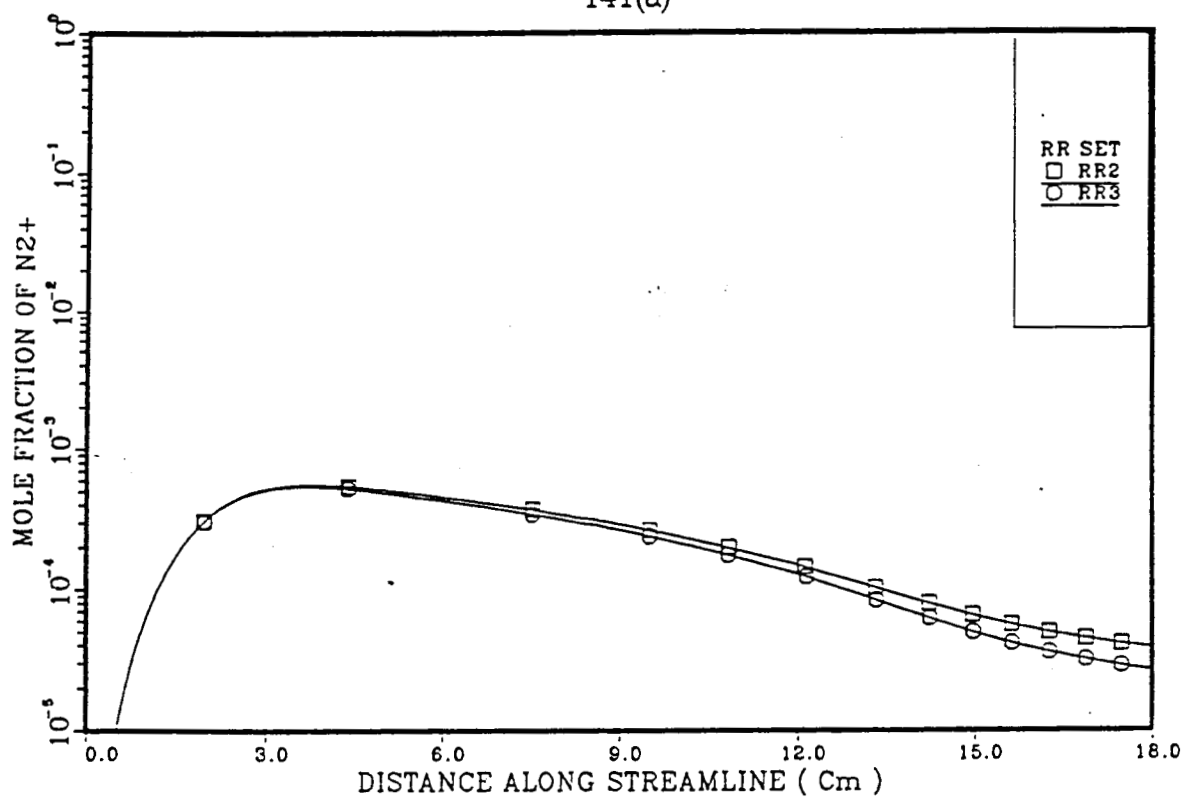


140(b)

FIGURES 140(a),140(b).PROFILES AT $V=7.7$ Km/s, PARK-L



141(a)



141(b)

FIGURES 141(a),141(b).PROFILES AT $V=7.7$ Km/s, PARK-L

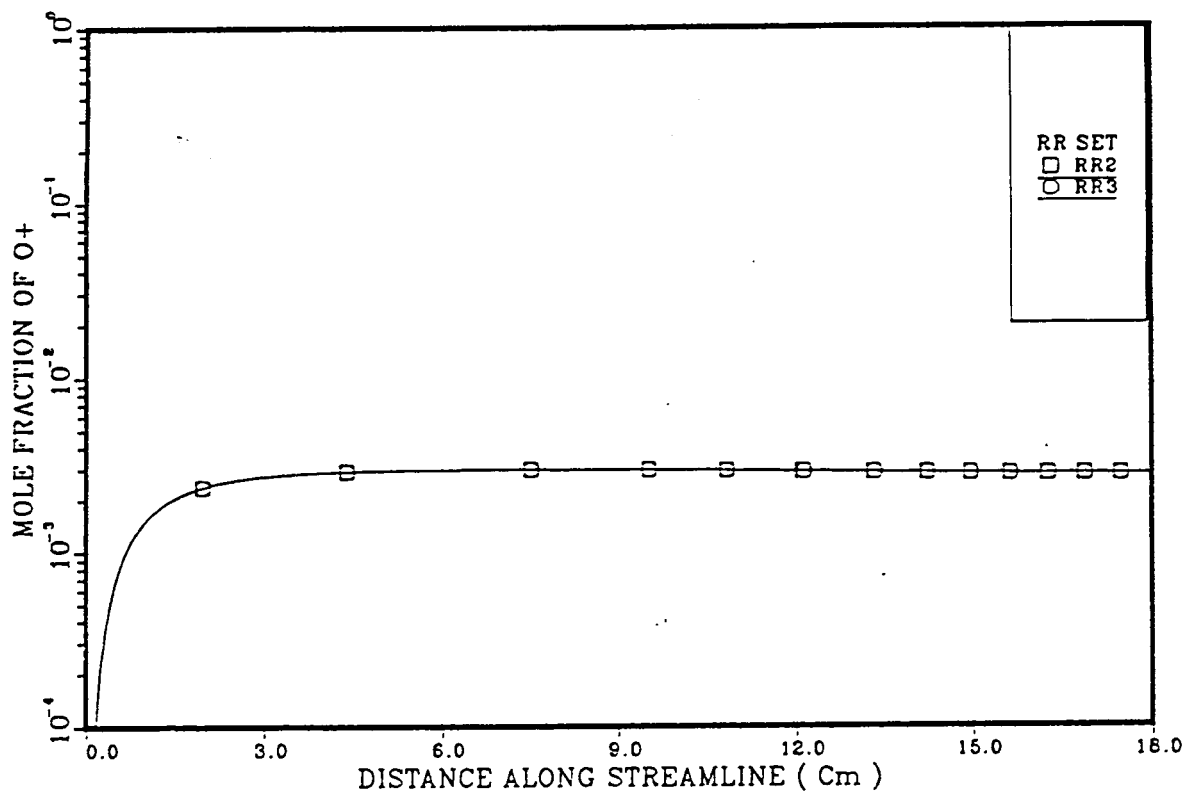


FIGURE 142.PROFILE AT V=7.7 Km/s, PARK-L

APPENDIX 4

Master's Thesis of Robert B. Greendyke

"A Parametric Study of Shock Jump Chemistry, Electron Temperature, and
Radiative Heat Transfer Models in Hypersonic Flows"

A PARAMETRIC STUDY OF SHOCK JUMP CHEMISTRY, ELECTRON
TEMPERATURE, AND RADIATIVE HEAT TRANSFER MODELS
IN HYPERSONIC FLOWS

A Thesis

by

ROBERT BRIAN GREENDYKE

Submitted to the Graduate College of
Texas A&M University
in partial fulfillment of the requirements for
the degree of

MASTER OF SCIENCE

August 1988

Major Subject: Aerospace Engineering

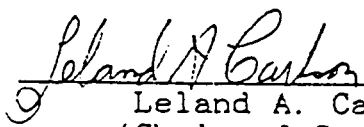
A PARAMETRIC STUDY OF SHOCK JUMP CHEMISTRY, ELECTRON
TEMPERATURE, AND RADIATIVE HEAT TRANSFER MODELS
IN HYPERSONIC FLOWS


A Thesis

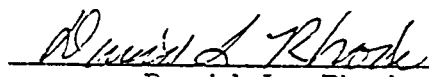
by

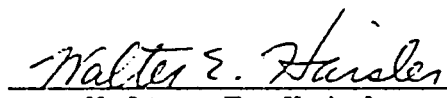
ROBERT BRIAN GREENDYKE

Approved as to style and content by:


Leland A. Carlson
(Chair of Committee)


Richard E. Thomas
(Member)


David L. Rhode
(Member)


Walter E. Haisler
(Head of Department)

August 1988

ABSTRACT

A Parametric Study of Shock Jump Chemistry, Electron
Temperature, and Radiative Heat Transfer Models
in Hypersonic Flows. (August 1988)

Robert Brian Greendyke, B.B.A., Baylor University
B.S., Texas A&M University

Chair of Advisory Committee: Dr. Leland A. Carlson

This thesis examines various engineering models of three physically important aspects of hypersonic flows. The thesis begins with the development of two computational models for simulating the chemical reactions across a hypersonic shock wave in air. In addition, two methods for the computation of the electron temperature and four methods of modelling the radiative heat transfer were incorporated into an existing modified inviscid flowfield solution method. In order to obtain accurate results from radiative heat transfer models, two correction factors for local thermodynamic nonequilibrium conditions were also developed. These three types of flow property models were subsequently combined with various chemical reaction rate and vibration - dissociation coupling models, and comparative solutions were obtained for the flowfield over an aeroassisted flight experiment vehicle.

DEDICATION

To my parents for their continual support of a
wayward son, and to Laina Walden for helping me to
remember.

ACKNOWLEDGMENTS

I wish to thank Dr. Leland Carlson for his guidance, support, and sense of humor throughout this research project. I would also like to thank Dr. Richard Thomas and Dr. David Rhode for their advice and for consenting to join my committee. In addition, I would like to thank Scott and Cindy Cullen for there help with the figures in this thesis.

This research was primarily supported by NASA Johnson Space Center under Grant No. NAG 9-192 with Dr. Carl Scott, Aerosciences Branch, as technical monitor. In addition, the author acknowledges the College of Engineering of Texas A&M University who generously provided computational support.

TABLE OF CONTENTS

	Page
INTRODUCTION.....	1
BASIC FLOWFIELD SOLUTION METHOD.....	7
FLOW PROPERTY MODELS.....	18
Development of the Shock Jump Chemistry Model.....	18
Electron Temperature Models.....	23
Radiative Heat Transfer Models.....	27
Nonequilibrium Correction Factors for Radiation.....	30
RESULTS AND DISCUSSION.....	38
Shock Jump Chemistry Models.....	39
Electron Temperature Models.....	47
Radiative Heat Transfer Models and Nonequilibrium Correction Factors.....	53
CONCLUSION.....	78
REFERENCES.....	81

LIST OF TABLES

	Page
Table 1. Step Model Band Frequencies.....	29
Table 2. Chemical Reaction Model.....	38
Table 3. Comparison of Computational Steps Taken by Flowfield Solution Method to Reach X = 10 cm.....	40
Table 4. Q_r for Wilson Rate and CVDV Coupling Model at X = 9.0 cm.....	65
Table 5. Q_r for Wilson Rate and CPL Coupling Model at X = 9.0 cm.....	66
Table 6. Q_r for Kang and Dunn Rate and CVDV Coupling Model at X = 9.0 cm.....	67
Table 7. Q_r for Kang and Dunn Rate and CPL Coupling Model at X = 9.0 cm.....	68

LIST OF FIGURES

	Page
Figure 1. Shock Coordinate System.....	8
Figure 2. Computational Step Sizes Along Streamline 4.....	41
Figure 3. Computational Step Sizes Along Streamline 6.....	42
Figure 4. Comparison of Heavy Particle Temperatures for Shock Jump Conditions 1 and 2 on Streamline 4.....	44
Figure 5. Comparison of O ₂ Concentrations for Shock Jump Conditions 1 and 2 on Streamline 4.....	45
Figure 6. Comparison of Atomic Oxygen Concentrations for Shock Jump Conditions 1 and 2 on Streamline 4.....	46
Figure 7. Carlson and Rieper T _e Correlation at 10 km./sec. for Streamline 4.....	49
Figure 8. Carlson and Rieper T _e Correlation at 7.5 km./sec. for Streamline 4.....	50
Figure 9. Park Assumption T _e at 10 km./sec. for Streamline 4 with CVDV Vibration- Dissociation Coupling Model.....	51

Figure 10. Park Assumption T_e at 10 km./sec. for Streamline 4 with CPL Vibration- Dissociation Coupling Model.....	52
Figure 11. Comparison of T_e for Wilson Reaction Rate at $X=9.0$ cm.....	54
Figure 12. Comparison of T_e for Kang and Dunn Reaction Rate at $X=9.0$ cm.....	55
Figure 13. Comparison of T_e for Sandia Reaction Rate at $X=9.0$ cm.....	56
Figure 14. Comparison of N_N for Wilson Reaction Rate at $X=9.0$ cm.....	57
Figure 15. Comparison of N_N for Kang and Dunn Reaction Rate at $X=9.0$ cm.....	58
Figure 16. Comparison of N_N for Sandia Reaction Rate at $X=9.0$ cm.....	59
Figure 17. Comparison of N_{N+} for Wilson Reaction Rate at $X=9.0$ cm.....	60
Figure 18. Comparison of N_{N+} for Kang and Dunn Reaction Rate at $X=9.0$ cm.....	61
Figure 19. Comparison of N_{N+} for Sandia Reaction Rate at $X=9.0$ cm.....	62
Figure 20. Nitrogen Correction Factors for Wilson Rate with CVDV Coupling.....	69

Figure 21. Nitrogen Correction Factors for Wilson Rate with CPL Coupling.....	70
Figure 22. Nitrogen Correction Factors for Kang and Dunn Rate with CVDV Coupling.....	71
Figure 23. Nitrogen Correction Factors for Kang and Dunn Rate with CPL Coupling.....	72
Figure 24. Incremental Radiative Heat Transfer to the Wall from Uncorrected CVDV Model Using Kang and Dunn Rate at X=9.0 cm.....	74
Figure 25. Incremental Radiative Heat Transfer to the Wall from Corrected CVDV Model Using Kang and Dunn Rate at X=9.0 cm.....	75
Figure 26. Incremental Radiative Heat Transfer to the Wall from Corrected CPL Model Using Kang and Dunn Rate at X=9.0 cm.....	76

NOMENCLATURE

B	excited state, or black body function
C, c	mass fraction, or speed of light
e	energy, or electron
E_2	exponential integral of type 2
f	correlation factor: $f=0$ for monotomic species, $f=1$ for diatomic species
g	degeneracy
ΔH	standard heat of formation at $0^\circ K$
h	static enthalpy
m	mass of a particle
N	number density, or nitrogen
O	oxygen
P, p	pressure, or excited state
Q, q	partition function, or excited state
Q_r, q_r	radiative heat transfer
R	universal gas constant
R_c	shock radius of curvature
r	shock coordinate (see Figure 1.)
S	rate of species production, or source function
T	temperature
t	time
u	velocity in x direction
v	velocity in y direction
X	ground state
x, y	shock coordinates (see Figure 1.)
1	ground state
α	degree of ionization
β	degree of dissociation
δ	variable defined by equation (31c)
ϵ	energy

Θ, θ	characteristic temperature, or local shock inclination angle
κ	absorption coefficient
Λ	variable defined by equation (31a)
λ	scale factor $(1-y)/R_o$
μ	molecular weight
ν	wavelength, or stoichiometric coefficient
ρ	density
σ	radiative cross-section
σ_v	variable defined by equation (22d)
τ	relaxation time, or optical thickness
ψ	stream function
Ω	variable defined by equation (31b)
ω_k	rate of production, or rate of change of vibrational energy of species k

Subscripts

a, av	average
b	body, or backward
d	dissociation
dn	dissociation of nitrogen
do	dissociation of oxygen
e	electron
f	forward
h	heavy particle (translational)
I	ionization
i	species
j	reaction
l, l	energy level
s	shock
v	vibrational
ν	wavelength
∞	freestream, or previously undissociated

INTRODUCTION

Several new aerospace engineering projects have brought about a renewed interest in hypersonic flows in the last few years. The recent decision by NASA to build a manned space station has led to the need for a transfer vehicle to travel back and forth between a low earth orbit, such as that used by the Space Shuttle, and the higher earth orbit that would be occupied by the Space Station. Such a vehicle could also be used to transport supplies and people between the Moon, or even Mars, and an earth orbit. Many engineers have proposed the development of an aero-assisted orbital transfer vehicle, or AOTV, to accomplish this task. In order for the AOTV to descend to a lower orbit, it would skim through the outer layers of the earth's atmosphere. The resulting loss in momentum due to the aerodynamic drag forces on the AOTV would allow it to descend to a lower orbit without the need to fire a retro-rocket. Consequently, less fuel would need to be carried onboard the AOTV; and cargo could be carried in place of the fuel. For such a vehicle to be successful, the need for fast and accurate computational design methods is critical. The most difficult problem in developing these methods is the complexity of modelling the physical phenomenon present at the hypersonic speeds used in the aerocapture of the AOTV.

An AOTV transferring from a geosynchronous earth orbit to a low earth orbit will typically enter the earth's atmosphere at a speed of 10 kilometers per second with a

Format in accordance with the AIAA Journal

maximum dynamic pressure occurring at an altitude of 75 to 80 kilometers. There are three physical processes that occur in this flow regime that are unique to hypersonic flowfields. The first process is the dissociation of the molecular nitrogen and oxygen present in the atmosphere, and their subsequent chemical reactions and recombinations into new chemical species. These chemical reactions have a strong effect on the flowfield, and the most important effect is the influence on the temperature of the flow. The dissociation of nitrogen and oxygen converts a large amount of the translational energy that the molecules acquire in traversing the strong bow shock wave into the dissociation process itself. Since translational temperature is the average energy random motion of the atoms and molecules in the air, the net result is a decrease in translational temperature immediately behind the shock wave. This loss in temperature and the corresponding chemical reactions have a strong influence on the pressure and density of the flow, and consequently, will have a large impact on the design of an AOTV.

The high speed and strong shock wave inherent in hypersonic flows also cause thermal nonequilibrium to develop between the translational mode of molecular energy and the other modes of energy, which attain equilibrium with one another by collisional processes. Due to their low mass, electrons are inefficient at reaching an equilibrium with the local translational, or heavy particle, temperature in a collision; and typically, the electron temperature (T_e), representing the energy of both bound and free electrons, will be lower than the translational temperature throughout most of the region of chemical nonequilibrium.

The third process unique to hypersonic flows is thermal radiation. As bound electrons change from a higher

energy state to a lower energy state, they emit the excess energy in the form of line radiation. Free electrons produced by ionization will similarly emit a continuum radiation, called Bremsstrahlung, from the conversion of translational energy or by deionization, and diatomic molecules emit band radiation associated with electronic transitions involving various vibrational states. In the flow regime of the AOTV, the radiative heat transported to the AOTV's heat shield could be a significant portion of the total heat transfer; and since much of the radiation is the result of electron state changes and free electrons deionizing into band states, the prediction of thermal radiation is highly dependant on the accurate determination of the electron temperature. The design of an AOTV heat shield will, in turn, be dependant on the accurate prediction of the radiative heat transfer.

Immediately behind a strong bow shock wave, the diatomic nitrogen and oxygen present will quickly dissociate. Other chemical species will form and react, but the additional chemical reactions will not occur as rapidly. Current computational methods of calculating chemical nonequilibrium require small computational steps in areas of rapid chemical change, and consequently, the more rapid the chemical changes, the more computational effort required to solve for the flowfield. Part of this thesis will attempt to find a method for quickly calculating the equilibrium chemical concentrations immediately behind the bow shock wave for oxygen and nitrogen which should eliminate the time consuming steps associated with the rapid dissociation processes. Obviously, modelling the shock chemistry in such a fashion is only valid for speeds at which the oxygen or nitrogen would have dissociated in a close spacial distance to the shock front. Since oxygen will typically dissociate close

to the shock at a much lower speed than will nitrogen, two models will be presented. The first will consider nitrogen concentrations to be frozen at the shock with oxygen concentrations being in equilibrium at the local translational temperature immediately behind the shock front. It is felt that this model will be valid in a velocity range of 7 to 10 kilometers per second. The second model will be one in which both oxygen and nitrogen concentrations will be in equilibrium with the local translational temperature of the shock front. This model will only be valid at very high speeds typical of a re-entry from Mars.

Based on past experience, there are several methods for the determination of the electron temperature. The first approach is to include in the flow model the highly coupled electron energy equation¹. However, the complexity and sensitivity of this equation makes the flow equations difficult to solve; and for many practical engineering applications this method is too cumbersome. A second method² is based on certain assumptions about the processes involved in the electronic excitation process. In this approach, the initial excitation is assumed to be dominated by atom-molecule collisions followed by a flow region where electronic excitation is strongly coupled to the level of vibrational excitation of diatomic nitrogen; and a correlation was developed based upon the chemical species present and the local translational or vibrational temperature. There has also been strong evidence that the electron temperature for both bound and free electrons is coupled to the vibrational temperature of diatomic nitrogen. An additional method of determining the electron temperature, therefore, is to simply let it equal the vibrational temperature of nitrogen³. Since a twenty percent change in the electron temperature can cause an

order of magnitude difference in the radiative heat transported to the AOTV body, it is vital that various electron temperature models be investigated in this study.

To model the radiative heat transfer, several options are available. The quickest and easiest model to use is an optically thin radiance model⁴. The radiance model is based on several radiative processes, and the amount of thermal radiation is determined by the local concentrations of the participating chemical species and the electron temperature. In this model, it is assumed that the air does not self-absorb the radiation emitted. However, in actuality, a significant amount of the emitted radiation is absorbed by the air. In order to account for this "non-gray" absorption, engineers have developed several step models that approximate the spectral dependance of the absorption characteristics of air. While such step models allow a quick determination of the emission and absorption behavior, they differ in what radiative effects are included. In addition, since these step models are usually based on factors such as local species concentrations, electron temperature, and local air density, they are quick and simple to use; but, unfortunately, the radiative heat transfers predicted by them can differ by as much as a factor of two. In principle, the radiative flux should be determined by integrating the spectrum of air in a line-by-line, band by band fashion; but this approach is usually too time consuming to be practical for most engineering applications. Thus, step models of absorption coefficients are necessary in most cases. Part of this study will examine the radiance model and various step models in order to determine their appropriateness to the flight regime of the AOTV.

The final area to be investigated will be the

effect of nonequilibrium corrections on the radiative heat transfer models studied. All of the approximate radiative transfer models assume local thermodynamic equilibrium (LTE), which requires that a Boltzmann distribution exist between the excited states and the ground state of the chemical species in the models. Under chemical equilibrium conditions, this assumption is valid. However, Chapin⁵ indicated that under conditions of chemical and thermal nonequilibrium certain correction factors might be necessary to account for any local thermodynamic nonequilibrium (LTNE) deviations from a Boltzmann distribution induced by chemical processes between excited states and the ions and electrons present in the flow. Uncorrected, radiation models could exaggerate the actual radiative transfer due to atomic processes. The effect of the ionization correction factors will be examined for each of the radiative heat transfer models studied. In addition, a new correction factor will be postulated that approximately accounts for deviations from a Boltzmann distribution of the electronic states of the diatomic molecules. Not including the effects of such deviations could also lead to significant overestimation of the molecular radiative heat transfer to an AOTV.

BASIC FLOWFIELD SOLUTION METHOD

A flowfield solution method for the basic AOTV flowfield had to be chosen before any of the various models of physical phenomena could be tested. Since uncoupled radiative heat transfer could be legitimately assumed for the speed regime of the AOTV, the choice of a solution method could be made independently of the radiation models. However, there were still several requirements for the flowfield solver. The method had to be adaptable to the geometry of a proposed AOTV. In addition, in order to obtain the proper flowfield properties for the radiative heat transfer and electron temperature models, the solution method had to include nonequilibrium chemistry, vibration - dissociation coupling models, and electronic excitation. The ability to study the chemical reactions along individual streamlines was also desired for the electron temperature models. Finally, as with all computational fluid dynamics methods, an efficient method was essential. Consequently, an inverse flowfield solution method first proposed by Maslen⁶, and later modified by Grose⁷, for solving the nonequilibrium inviscid flowfield about blunt hypersonic vehicles was selected as the basic computer code for this project.

With a strong, detached shock wave, it is possible to define a shock-oriented, curvilinear coordinate system (Figure 1). After carrying out a Von Mises transformation on the coordinate system, the independent variables are x , the distance along the shock, and ψ , the stream function. In this transformed coordinate system, the continuity equation becomes

$$\frac{\partial}{\partial x}(\rho u r) + \frac{\partial}{\partial y}(\rho v \lambda r) = 0 \quad (1)$$

from which the stream function can be defined as

$$\frac{\partial \psi}{\partial x} = \rho v r \lambda \quad (2)$$

and,

$$\frac{\partial \psi}{\partial y} = -\rho u r \quad (3)$$

The momentum equations are

$$\left[\frac{\partial u}{\partial x} \right]_{\psi} - \frac{v}{R_c} + \frac{\lambda v r}{u} \left[\frac{\partial p}{\partial \psi} \right]_x + \frac{1}{\rho u} \left[\frac{\partial p}{\partial x} \right]_{\psi} = 0 \quad (4)$$

and,

$$\left[\frac{\partial v}{\partial x} \right]_{\psi} + \frac{u}{R_c} - \lambda r \left[\frac{\partial p}{\partial \psi} \right]_x = 0 \quad (5)$$

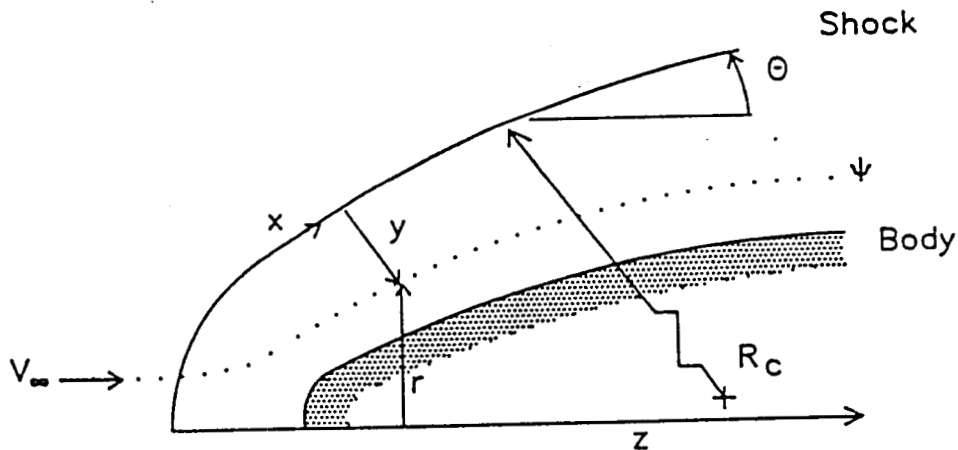


Figure 1. Shock Coordinate System.

and the energy equation is

$$\left[\frac{\partial h}{\partial x} \right]_{\psi} = \frac{1}{\rho} \left[\frac{\partial p}{\partial x} \right]_{\psi} \quad (6)$$

Two other equations are necessary to fully specify the flowfield. These are the state equation

$$h = h(p, \rho, c_1, c_2, \dots, c_k) \quad (7)$$

and the rate equation

$$\left[\frac{\partial c_k}{\partial x} \right]_{\psi} = \frac{\lambda \omega_k}{u} \quad (8)$$

Since the flow in an inviscid shock layer is nearly parallel to the shock, it can be assumed that the

$\left[\frac{\partial v}{\partial x} \right]_{\psi}$ is zero; and with the additional assumption that λ is approximately one, equation (5) becomes

$$\left[\frac{\partial p}{\partial \psi} \right]_x = \frac{u}{R_c r} \quad (9)$$

In equation (9), Maslen approximated the values for r and u by the values immediately behind the shock; and with this approximation, equation (9) can be integrated to yield

$$p(x, \psi) = p_s(x) + \frac{u_s(x)}{R_c(x) r_s(x)} [\psi - \psi_s(x)] \quad (10)$$

where, at the shock

$$\psi_s = \frac{\rho_{\infty} V_{\infty}^2 r_s^2}{2} \quad (11)$$

and, on the body

$$\psi_b = 0 \quad (12)$$

With this approach, equation (10) is thereby

decoupled from the solution of the other conservation equations and can be solved independently of them. Notice that the values of the flow properties at the shock can be easily obtained for a known shock geometry and proper assumptions concerning the shock jump chemistry.

Since the flow closely parallels the shock at hypersonic velocities, it can further be assumed that

$$v^2 - v_o^2 \ll u^2 \quad (13)$$

The conservation of energy

$$h + \frac{u^2}{2} + \frac{v^2}{2} = h_o + \frac{v_o^2}{2} \quad (14)$$

can then be used to solve for the velocity

$$u = \left[2(h_o - h) \right]^{1/2} \quad (15)$$

Grose's modification of Maslen's method allowed the internal energy of the gas to include the effects of translational, rotational, and vibrational motion, as well as the effects of dissociation, electronic excitation, and ionization of the chemical species. Accordingly, the internal energy of the i th species is

$$e_i = \frac{3}{2} \frac{RT}{\mu_i} + f_i \frac{RT}{\mu_i} + \epsilon_{v,i} + \frac{R}{\mu_i} \frac{\sum_{l=1}^{L_i} g_{i,l} \epsilon_{i,l} \exp\left\{ \frac{-\epsilon_{i,l}}{T} \right\}}{\sum_{l=1}^{L_i} g_{i,l} \exp\left\{ \frac{-\epsilon_{i,l}}{T} \right\}} + \frac{\Delta H_i}{\mu_i} \quad (16)$$

In Grose's original computer code the vibrational energy was found by one of two methods at the user's discretion. The first option assumed vibrational motion to be in equilibrium with the translational mode of motion, and

for this case the vibrational energy is simply

$$\epsilon_{v,i} = \frac{f_i R \Theta_i}{\mu_i \left[\exp\left(\frac{\Theta_i}{T}\right) - 1 \right]} \quad (17)$$

The second option, a Coupled Vibration - Dissociation (CVD) model⁸, assumes that vibrational energy is not in equilibrium with the translational energy. In the CVD model, vibrational energy can be represented by a system of simple harmonic oscillators where the rate of change of vibrational energy is

$$\frac{\partial \epsilon_{v,i}}{\partial t} = \frac{\epsilon_{v,i\infty} - \epsilon_{v,i}}{\tau} \quad (18)$$

The CVD model then relates the effects of vibrational nonequilibrium to dissociation by the use of a coupling coefficient, which is a correction factor to the forward chemical rate that reflects the difference between the actual forward rate and the local forward rate⁹ if vibrational equilibrium existed.

In a related work, Bobskill⁹ added three additional vibrational nonequilibrium models to the Grose computer code. The first of these is the Coupled Vibration - Dissociation - Vibration (CVDV) model⁴ where the rate of change of vibrational energy is represented by

$$\frac{\partial \epsilon_{v,i}}{\partial t} = \frac{\epsilon_{v,i\infty} - \epsilon_{v,i}}{\tau} \cdot \left\{ \left[\frac{\Theta_{v,i}}{\exp\left(\frac{\Theta_{v,i}}{T_m}\right) - 1} - \frac{N\Theta_{v,i}}{\exp\left(\frac{N\Theta_{v,i}}{T_m}\right) - 1} \right]^{-\epsilon_{v,i}} \right\} \\ + \frac{1}{n} \left[\frac{\partial n}{\partial t} \right]_f + \left[\frac{1}{2} (N - 1) \Theta_{v,i} - \epsilon_{v,i} \right] \frac{1}{n} \left[\frac{\partial n}{\partial t} \right]_r \quad (19)$$

The first term in equation (19) is the vibrational rate term found in the CVD model. The second term represents the average energy lost by the dissociation of a single molecule, while the third term represents the vibrational energy gained through the recombination of a molecule. The coupling factor in the CVD model is again used to relate the effects of vibrational nonequilibrium on the dissociation reactions.

The second vibrational nonequilibrium model added by Bobskill was the CVDV - Preferential model⁴. In the CVDV - Preferential model, it was assumed that dissociation preferentially occurs from the higher energy vibrational levels. By contrast, the CVD and CVDV models assumed that dissociation can occur with equal probability from any vibrational level. In addition, the CVDV - Preferential model assumes that vibrational energy can be represented by a system of anharmonic oscillators, while the CVD and CVDV models assumed a system of simple harmonic oscillators. The rate of change of vibrational energy in the CVDV - Preferential model is then

$$\begin{aligned} \frac{\partial \epsilon_{v,i}}{\partial t} = & \frac{\epsilon_{v,i\infty} - \epsilon_{v,i}}{\tau} + \left[\frac{\bar{E}(T, T_{v,i}) - \epsilon_{v,i}}{n} \right] \left[\frac{\partial n}{\partial t} \right]_f \\ & - \left[\frac{\bar{G}(T) - \epsilon_{v,i}}{n} \right] \left[\frac{\partial n}{\partial t} \right]_r \end{aligned} \quad (20)$$

The final vibrational nonequilibrium model added by Bobskill was the two temperature chemical-kinetic "Chul Park - Like" (CPL) model³. The CPL model assumes that the effective electron temperatures are equal to the nitrogen vibrational temperature, while the rotational temperature is in equilibrium with the translational temperature. The rate of vibrational nonequilibrium can then be represented

by

$$\begin{aligned} \frac{\partial \epsilon_{v,i}}{\partial t} = & \frac{[\epsilon_{v,i\omega} - \epsilon_{v,i}]}{\tau_l} \left[\frac{T - T_{v,i}}{T_s - T_{v,i}} \right]^{(S-1)} \\ & - \left\{ \left[\frac{\Theta_{v,i}}{\exp\left(\frac{\Theta_{v,i}}{T_m}\right) - 1} - \frac{N\Theta_{v,i}}{\exp\left(\frac{N\Theta_{v,i}}{T_m}\right) - 1} \right] - \epsilon_{v,i} \right\} \\ & \frac{1}{n} \left[\frac{\partial n}{\partial t} \right]_f + \left[\frac{1}{2} (N - 1) \Theta_{v,i} - \epsilon_{v,i} \right] \frac{1}{n} \left[\frac{\partial n}{\partial t} \right]_r \end{aligned} \quad (21)$$

where

$$S = 3.5 e^{(-5000/T_s)} \quad (22a)$$

$$\tau_l = \tau + \tau_c \quad (22b)$$

$$\tau_c = 1 / (n C \sigma_v) \quad (22c)$$

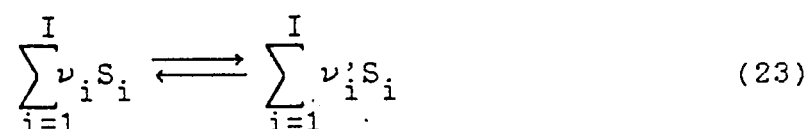
$$\sigma_v = 10^{-17} (50000/T)^2 \quad (22d)$$

The rate of change in equation (21) has an additional multiplier on the first term that accounts for the nonlinear variation in the vibrational relaxation rate with the difference in vibrational energies. The CPL model further assumes that there is a limiting cross section (σ_v) at the high temperatures in the shock front. The equations in (22) are correction factors to the linear assumption. In the original Park model³, it was assumed that the temperatures used to compute the forward dissociation reactions should be a geometric average of the translational temperature and the vibrational temperature:

$$T_a = \sqrt{TT_v} \quad (22)$$

In this way, the forward rates are properly reduced due to vibration - dissociation coupling. However, in the present CPL model, equation (22) is not used because it is very approximate and the previously discussed coupling coefficient is used instead. In addition, in the actual Park model a combined electron - vibrational energy equation is used which not only accounts for vibrational and electron/electronic energy changes but also includes vibrational - electronic coupling. However, in the present model it is assumed that electron/electronic effects on vibrational energy are small, but that the nitrogen vibrational effects on the effective electron temperature are very strong. With this assumed one-way coupling, the nitrogen vibrational energy and temperature can be computed using equation (21) and the electron temperature is assumed equal to T_{vN2} .

In addition to modelling the vibrational energy, it is also necessary to model the chemical reactions. Following the discussion of Grose⁷, an arbitrary chemical reaction can be represented as



By the law of mass action, the rate of production of species S_i is in direct proportion to the product of the concentration of each species raised to power of its stoichiometric coefficient. The rate of production of a species S_i , per unit mass, is then

$$\frac{d\left[\frac{S_i}{\rho}\right]}{dt} = \frac{K}{\rho} \left(\nu'_i - \nu_i \right) \prod_{i=1}^I \left(S_i \right)^{\nu_i} \quad (24)$$

In Grose's coordinate system, and noting that

$$\left[S_i \right] = \frac{\rho c_i}{\mu_i} \quad (25)$$

equation (24) can then be represented as

$$\left[\left[\frac{\partial c_i}{\partial x} \right]_{\psi} \right]_j = \frac{K_j \mu_i}{\rho u} \left(\nu'_{i,j} - \nu_{i,j} \right) \prod_{i=1}^I \left(\frac{\rho c_i}{\mu_i} \right)^{\nu_{i,j}} \quad (26)$$

where

$$K_j = A_j T^{B_j} \exp \left(\frac{-E_j}{T} \right) \quad (27)$$

The vibrational coupling factor⁹ (ϕ_j) is simply multiplied on the right hand side of equation (26) to incorporate the effects of vibrational nonequilibrium

$$\left[\left[\frac{\partial c_i}{\partial x} \right]_{\psi} \right]_j = \frac{\phi_j K_j \mu_i}{\rho u} \left(\nu'_{i,j} - \nu_{i,j} \right) \prod_{i=1}^I \left(\frac{\rho c_i}{\mu_i} \right)^{\nu_{i,j}} \quad (28)$$

Since the Grose method is an inverse one, the first step in the general solution procedure is the specification of the shock shape, where

$$r_s = r_s(z) \quad (29)$$

The local shock inclination angle (θ), the radius of curvature (R_c), and the x - direction spacial coordinate can then be found from

$$\theta = \arctan \left[\frac{dr_s}{dz} \right] \quad (30a)$$

$$R_c = \frac{\left[1 + \left(\frac{dr_s}{dz}\right)^2\right]^{3/2}}{\left|\frac{d^2 r_s}{dz^2}\right|} \quad (30b)$$

$$x = \int_0^{r_s} \frac{dr_s}{\sin \theta} \quad (30c)$$

With the freestream conditions specified by user input, the next step in the solution procedure is the determination of the translational temperature immediately behind the shock (T_s). In order to accomplish this task, Grose specified the three oblique shock conservation relations by

$$\Lambda(x) = \rho_\infty V_\infty \sin(\theta) \quad (31a)$$

for the conservation of mass, and

$$\Omega(x) = p_\infty + \rho_\infty V_\infty^2 \sin^2(\theta) \quad (31b)$$

for the conservation of momentum, and finally

$$\delta(x) = e_\infty + \frac{p_\infty}{\rho_\infty} + \frac{V_\infty^2 \sin^2(\theta)}{2} \quad (31c)$$

for the conservation of energy. With an initial guess of T_s made by the Rankine - Hugoniot relations, the internal energy at the shock (e_s) can be found from equation (16). The second, and subsequent, iterations for T_s are then determined by

$$T_s = \frac{2\mu_\infty(\delta - e_s)\sqrt{\Omega^2 - 2\Lambda^2(\delta - e_s)}}{R\left[\Omega + \sqrt{\Omega^2 - 2\Lambda^2(\delta - e_s)}\right]} \quad (32)$$

where equation (32) represents all three shock conservation equations solved for T_s . The requirement that enthalpy remains constant along a streamline is used as the basis for the iterations. Once the value of T_s differs from the previous value by a user - specified tolerance, the remaining shock parameters can be found from

$$p_s = \sqrt{\Omega^2 - 2\Lambda^2(\delta - e_s)} \quad (33a)$$

$$\rho_s = \frac{p_s \mu_\infty}{RT_s} \quad (33b)$$

$$h_s = e_s + \frac{p_s}{\rho_s} \quad (33c)$$

$$u_s = V_\infty \cos(\theta) \quad (33d)$$

The determination of T_s will be modified later in this study to incorporate a proposed shock jump chemistry model.

The pressure distribution in the Von Mises plane can then be obtained from equation (10). With the pressure distribution known, the chemical rate and vibrational energy equations can be integrated along with the equations for flow properties along streamlines to find h , c_i , $\epsilon_{v,i}$, ρ , and T . The complete flowfield is then determined.

FLOW PROPERTY MODELS

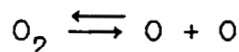
Development of the Shock Jump Chemistry Model

In the region of nonequilibrium chemistry behind the shock wave of an AOTV transferring from high to low Earth orbit, the diatomic oxygen present in Earth's atmosphere will quickly dissociate into atomic oxygen. At the velocities during the initial re-entry phase of an AOTV returning from Mars, both the diatomic oxygen and the diatomic nitrogen in the atmosphere will similarly dissociate immediately behind the shock front. Most computational methods for solving a hypersonic flowfield rely on the fact that total enthalpy remains constant along a streamline. Consequently, in order to maintain this requirement in regions of rapid chemical changes, computational methods must, to preserve accuracy, reduce the spatial distance of each computational step. The net result of this rapid dissociation of diatomic oxygen and diatomic nitrogen at high velocities is a significant increase in the computational time necessary to calculate the flowfield in the chemical nonequilibrium region.

In order to possibly decrease the time necessary to calculate the flowfield of an AOTV, this study investigated the use of a model that will solve for the chemical concentrations and temperature at the shock front if oxygen, or both oxygen and nitrogen, were in chemical equilibrium at the translational temperature immediately behind the shock front. Of course, this model had to preserve conservation of mass, momentum, and energy. Two models were developed, and the first of these calculates the equilibrium concentration of atomic and diatomic oxygen

at the shock front while keeping the concentration of diatomic nitrogen frozen. The second model calculates the equilibrium concentrations of atomic and diatomic oxygen as well as the equilibrium concentrations of atomic and diatomic nitrogen at the shock front.

The first model was developed on the assumption that the only dissociative reaction for oxygen is



Then, by the law of mass action for a diatomic gas¹⁰

$$\frac{\beta^2}{1 - \beta} = \frac{1}{2N_{\text{O}\infty}} \frac{[Q_0]^2}{Q_{\text{O}_2}} \exp\left[-\theta_d/T\right] \quad (34)$$

where β is defined as the ratio of the number density of dissociated oxygen atoms to the number density of previously undissociated oxygen atoms, and $N_{\text{O}\infty}$ is defined as the number density of the previously undissociated oxygen atoms. Since the concentration of nitrogen is frozen at the shock front, total density at the shock can be defined as

$$\rho = \rho_{\text{N}_2} + \rho_{\text{O}} + \rho_{\text{O}_2} \quad (35)$$

or, by rearranging

$$1 - C_{\text{N}_2\infty} = \frac{\rho_{\text{O}_2}}{\rho} + \frac{\rho_{\text{O}}}{\rho} \quad (36)$$

and, noting that

$$N_{\text{O}\infty} = N_{\text{O}} + 2N_{\text{O}_2} \quad (37)$$

equation (36) can be further rearranged as

$$\frac{\rho V}{m_O} \left(1 - C_{N2\infty} \right) = N_{O\infty} \quad (38)$$

Substituting this result into equation (34) yields

$$\frac{\beta^2}{1 - \beta} = \frac{m_O}{2\rho V} \left(1 - C_{N2\infty} \right) \left(\frac{Q_O}{Q_{O2}} \right)^2 \exp \left(-\theta_d/T \right) \quad (39)$$

By noting that

$$\rho_s = \frac{P_s \mu_\infty}{(1+\beta)RT_s} \quad (40)$$

equation (39) then becomes

$$\begin{aligned} \frac{1 - \beta}{\beta^2} &= \frac{2P_s \mu_\infty}{(1+\beta)RT_s} \left(1 - C_{N2\infty} \right) \left(\frac{h^2}{\pi m_O k} \right)^{3/2} \left(T_s \right)^{-3/2} \\ &\times Q_{O2}^{\text{rot}} Q_{O2}^{\text{vib}} \frac{Q_{O2}^{\text{el}}}{\left(Q_O^{\text{el}} \right)^2} \exp \left(\theta_d/T_s \right) \end{aligned} \quad (41)$$

With initial estimates of T_s and P_s made from the Rankine - Hugoniot relations at the shock, the quadratic formula can be used to solve equation (41) for β . With β determined and the freestream concentrations known, the mass fractions for atomic and diatomic oxygen are

$$C_O = \beta C_{O2\infty} \quad (42a)$$

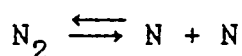
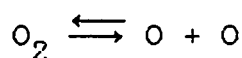
$$C_{O2} = (1 - \beta) C_{O2\infty} \quad (42b)$$

These concentrations can then be used to determine the internal energy

$$e_s = \sum e_{s_i} C_{s_i} \quad (43)$$

The e_s subsequently determines the enthalpy, and equations (32) and (33) can be used in the Grose code to make another estimate of T_s . Equations (41) and (42) are in turn used to make new estimates of the atomic and diatomic oxygen concentrations, and the whole process is repeated until T_s differs by a small amount from the previous iteration.

The determination of the equilibrium concentrations of oxygen and nitrogen at the shock follows a similar pattern. The assumption is made that two independent chemical reactions were responsible for the dissociation of oxygen and nitrogen



Then, at the shock, the density is

$$\rho = \rho_N + \rho_{N_2} + \rho_O + \rho_{O_2} \quad (44)$$

or,

$$\rho [1 - (C_N + C_{N_2})] = \rho_O + \rho_{O_2} \quad (45)$$

and, further simplification of the oxygen dissociation process leads to

$$\frac{\rho V}{m_O} [1 - C_{N_2\infty}] = N_{O\infty} \quad (46)$$

and, for nitrogen dissociation,

$$\frac{\rho V}{m_N} [1 - C_{O_2\infty}] = N_{N\infty} \quad (47)$$

Using the law of mass action for each independent chemical reaction yields

$$\frac{1 - \beta_O}{\beta_O^2} = 2N_{O\infty} \frac{Q_{O2}}{[Q_O]^2} \exp\left(\theta_{do}/T_s\right) \quad (48)$$

for oxygen, and

$$\frac{1 - \beta_N}{\beta_N^2} = 2N_{N\infty} \frac{Q_{N2}}{[Q_N]^2} \exp\left(\theta_{dn}/T_s\right) \quad (49)$$

for nitrogen.

As in the first shock jump chemistry model, equations (48) and (49) can be further reduced to

$$\begin{aligned} \frac{1 - \beta_O}{\beta_O^2} &= \frac{2P_s \mu_\infty}{(1 + \beta_O + \beta_N) RT_s} \left[1 - c_{N2\infty}\right] \left[\frac{h^2}{\pi m_O k}\right]^{3/2} T_s^{-3/2} \\ &\times Q_{O2}^{\text{rot}} Q_{O2}^{\text{vib}} \frac{Q_{O2}^{\text{el}}}{[Q_O^{\text{el}}]^2} \exp\left(\theta_{do}/T_s\right) \end{aligned} \quad (50a)$$

for oxygen, and

$$\begin{aligned} \frac{1 - \beta_N}{\beta_N^2} &= \frac{2P_s \mu_\infty}{(1 + \beta_O + \beta_N) RT_s} \left[1 - c_{O2\infty}\right] \left[\frac{h^2}{\pi m_N k}\right]^{3/2} T_s^{-3/2} \\ &\times Q_{N2}^{\text{rot}} Q_{N2}^{\text{vib}} \frac{Q_{N2}^{\text{el}}}{[Q_N^{\text{el}}]^2} \exp\left(\theta_{dn}/T_s\right) \end{aligned} \quad (50b)$$

for nitrogen. Equations (50) are a system of two independent nonlinear equations of two unknowns (β_O and β_N) that can be solved by a Newton - Raphson iteration with estimates of T_s and P_s . The corresponding mass fractions are

$$C_O = \beta_O C_{O2\infty} \quad (51a)$$

$$C_{O_2} = (1-\beta_O)C_{O_2\infty} \quad (51b)$$

$$C_N = \beta_N C_{N_2\infty} \quad (51c)$$

$$C_{N_2} = (1-\beta_N)C_{N_2\infty} \quad (51d)$$

which can be used to calculate the internal energy. The same iteration process as in the first model is used to determine the T_s .

It is very important to note that the shock jump chemistry models are only valid in certain ranges of freestream velocity. The first model is only expected to be valid for velocities around 10 km/sec, while the second model should be valid for velocities above 12-13 km/sec.

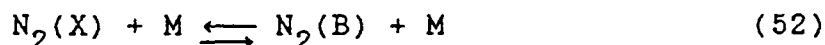
Electron Temperature Models

In the region of chemical nonequilibrium immediately behind a strong shock, thermal nonequilibrium develops between the translational mode of energy of the atoms or molecules and the translational energy of the electrons. Due to their low mass relative to that atoms, collisions are an inefficient process for equilibrating electrons with the heavy particles. In principle, the full electron energy equation can be used to solve for the electron temperature, but the electron energy equation is highly coupled and very time consuming to solve. Unfortunately, since radiation is governed by the electron temperature, the accurate determination of the electron temperature is essential for accurate radiative heat transfer predictions. For this study, two types of electron temperature models were investigated for the AOTV flowfield.

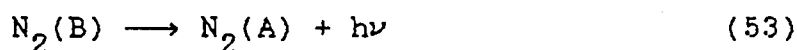
The first electron temperature model was developed by Carlson and Rieper², and is essentially a

correlation between experimental radiation data and a chemical reaction model. In the Carlson and Rieper model it was assumed that radiation in the $N_2(1+)$ band was the result of two reactions involving excited states. In the first step, a nitrogen atom is collisionally excited from the ground (or X state) to the A state, which was assumed to be in equilibrium with the B state of the nitrogen molecule. The molecules in the B state then radiate down to the A state.

The initial excitation can be expressed by the reaction



while the radiative emission is governed by



The rate of change of the B state can be expressed as

$$\frac{d[B]}{dt} = k_{f_{52}}[N_2(X)][M] - k_{f_{52}}[N_2(B)][M] - k_{53}[N_2(B)] \quad (54)$$

From the law of mass action, equations (53) and (54) become

$$\frac{[N_2(B)]}{[N_2(X)]} = \left(\frac{Q_B}{Q_X} \right) \exp(-E_{BX}/kT_e) \quad (55)$$

$$\frac{k_{f_{52}}}{k_{r_{52}}} = \left(\frac{Q_B}{Q_X} \right) \exp(-E_{BX}/kT_i) \quad (56)$$

With the assumption that the B state is in local equilibrium, the rate of change of the B state is

$$\frac{d[B]}{dt} = 0 \quad (57)$$

Carlson and Rieper then rearranged equation (54) to give

$$T_e = \frac{T_i}{1 + (kT_i/E_{BX}) \ln \left[1 + Q_B/Q_X / (K[M])^{-1} \right]} \quad (58)$$

where

$$K = C/k_{53} \quad (59)$$

and C and k_{53} were experimentally determined by Carlson and Rieper. The initial excitation of the $N_2(1+)$ radiative emission was assumed to be dominated by atom - molecule collisions. Therefore, the first collision partner is

$$[M] = [O] + [N]$$

and T_i becomes the translational temperature. Equation (58) then becomes

$$T_e = \frac{T}{1 + (T/85000) \ln \left[1 + \{10.1/([N]+[O])K\} \right]} \quad (60)$$

and

$$K = 4.23 \times 10^{-6} T^{-2.88} \text{ cm}^3 \quad (61)$$

However, after the vibrational temperature has peaked, the excitation process is assumed to be governed by vibrational - electronic coupling. The corresponding T_e can then be predicted through equation (58) by

$$T_e = \frac{T_{v_{N2}}}{1 + (T_{v_{N2}}/85000) \ln \left[1 + (10.1/[N_2]K_v) \right]} \quad (62)$$

and

$$K_v = 6.4 \times 10^{59} T_{v_{N2}}^{-19} \text{ cm}^3 \quad (63)$$

It is important to note that the correlations developed by

Carlson and Rieper were derived from data obtained in a shock tube for velocities at 6.4 km/sec.

Incorporating the Carlson and Rieper model into the Grose computer code proved to be relatively simple since the Grose code already calculated translational and vibrational temperatures along a streamline. All that was necessary to use the Carlson and Rieper model was to determine the number density of atomic oxygen and nitrogen and the number density of diatomic nitrogen. The number densities along a streamline were found by using the already determined density and mass fractions by

$$N_i = \frac{\rho N_{av} C_i}{\mu_i} \quad (64)$$

where N is the number density, and N_{av} is Avagadro's number.

The second model used in this study is, in reality, more of an assumption. Park³ has stated that the energy equilibration between the translational energy of free electrons and the vibrational motion of diatomic nitrogen is very rapid. In addition to Park, Lee^{1,11} found the same to be true. With the rapid equilibration of the temperature characteristic of the free electrons to the vibrational temperature of diatomic nitrogen and with the assumption that free and bound electrons are characterized by the same temperature, the vibrational temperature of diatomic nitrogen can be used as the electron temperature when solving for the radiative heat transfer. Since the Grose method already determined the vibrational temperature of nitrogen, the vibrational temperature information only had to be retained in order to use this assumption. It should be noted that in the present study it is inherently assumed that the vibrational - electronic coupling is essentially one way. In other words, vibrational processes

strongly influence the electron temperature but the electrons only have a small influence on the vibrational energy. It is believed that this equilibration of T_e and T_{vN_2} will be valid as long as there is a reasonable amount of nitrogen molecules in the flow.

Radiative Heat Transfer Models

For this study, four radiative heat transfer models were examined. One of these models is an optically thin radiance model, while the other three are non-gray step models. In the radiance model⁴, the air was assumed to be non-absorbing of the radiation it emitted by it. A set of rate equations was then fit to the predictions of Nardone et al¹² to calculate the radiation emitted by a small volume of gas. The rate equations are of the form

$$\text{Rad} = A_k N_j N_l \exp(-B_k/T_e) \quad (65)$$

where Rad is the radiative energy emitted per steradian by the kth radiation process, and A_k and B_k are constants. The original radiance model contained equations for twelve of these radiative processes, and this study incorporated a thirteenth process for the radiation in the $N_2^+(1-)$ band¹³. To incorporate the radiance model into the Grose code it was necessary to retain the number density and electron temperature data over the flowfield. The data was then used to calculate the radiation by the radiance equations and the radiation was integrated over space to obtain the total radiative flux to the body.

The non-gray step models are approximate representations of the self - absorption characteristics of air, and the steps model the variation of the absorption coefficient with wavelength over the total spectrum. The

non-gray step models represent an increase in the sophistication of radiative heat transfer modelling over the radiance model in that much of the radiation emitted by a gas, particularly in the ultraviolet region, is self-absorbed by the gas, and the radiance model can over-state the radiative flux to the surface of an AOTV.

The first step model used in the Grose code was developed by Olstad¹⁴. It contains eight steps which represent various portions of the ultraviolet, visible, and infrared regions of the electromagnetic spectrum (Table 1); and it includes effects due to atomic and molecular processes of both nitrogen and oxygen. One of the steps, step 7, is a "picket fence" representation of the ultraviolet line radiation; and another step, step 6, represents the continuum background and effects due to the line broadening in the same region. As shall be seen later, there exists the possibility that this step is slightly incorrect due to the fact that the Olstad model was derived at higher densities than those present in an AOTV flowfield and that the radiative cross-sections used are too large. While no temperature range was given for the Olstad model, a velocity regime of 10.7 to 15.2 km/sec at an altitude of 61 km was used by Olstad, and the AOTV flowfield should fall within the region of applicability of the Olstad model.

The second step model (KCN) used was developed by Knott, Carlson, and Nerem¹⁵. The KCN model was originally derived from experimental data for pure nitrogen and correlated to air. Since the radiative emission in air is dominated by nitrogen, it was felt that this was a valid approximation. The model consists of five bands and is applicable to air temperatures from the 8,000 to 20,000 °K.

The final step model used was that of Anderson¹⁶, and it consists of two steps. The first step accounts for

<u>MODEL/BAND</u>	<u>FREQUENCY(angstroms)</u>
Olstad	
Band	
1	400 - 852
2	852 - 911
3	911 - 1020
4	1020 - 1130
5	1800 - 4000
6	1130 - 1800
7	1130 - 1800 Lines
8	4000 - ∞
KCN	
Band	
1	620 - 1100
2	1100 - 1300
3	1300 - 1570
4	1570 - 7870
5	7870 - 9552
Anderson	
Band	
1	0 - 1100
2	1100 - ∞

Table 1. Step Model Band Frequencies

absorption behavior in the vacuum ultraviolet region of the spectrum, while the second step represents the absorption in the visible and infrared. The two step model is quick and easy to use, but the model was derived for high temperature equilibrium air in the region of 3,000 to 15,000 °K. The applicability of the Anderson model to a nonequilibrium AOTV flowfield is subject to some question since the temperatures near the shock at 10 km/sec approach

35,000 °K.

The tangent slab approximation was made for all of the radiative heat transfer models used in this study. According to the tangent slab approximation, the radiative flux to the surface is determined by the flow properties along a line perpendicular to the surface of the AOTV. Along with the tangent slab approximation, the assumption was made that the surface of the AOTV was non-emitting and that no precursor effects were present in the flowfield. Accordingly, the total radiative heat transferred to the surface of the AOTV can be represented by

$$q_r = 2\pi \int_0^\infty \int_0^S S(y) E_2[\tau(y)] \kappa(y) dy dv \quad (66)$$

Each of the radiative heat transfer models was incorporated into the Grose code in separate subroutines. All data necessary for their use was retained in the main part of the Grose code, and the individual radiation models called at the user's discretion.

Nonequilibrium Correction Factors For Radiation

The preceding radiative heat transfer models were all derived with the assumption that the electronic states of the radiating particles were populated according to a Boltzmann distribution at the local T_e . In a Boltzmann distribution, the local population of a given energy state (N_1^*) can be described by

$$N_1^* = N \frac{g_1 \exp(-\epsilon_1/kT)}{Q} \quad (67)$$

However, there is evidence⁵ that nitrogen and oxygen

ionization is a two step process involving excitation to an excited state followed by rapid ionization, in which it can be assumed that the excited states are in equilibrium with the electrons and ions at the local T_e and pressure. Consequently, the radiative heat transfer models need to be corrected for the fact that the electronic states are not that predicted by a Boltzmann distribution assuming local thermodynamic equilibrium. It should be noted that the following correction factors are only engineering approximations of the actual phenomena.

Atomic Correction Factors

To derive the correction factors for radiative processes involving atoms, it is necessary to consider four energy level transitions. The first is the transition of an electron from the continuum state to an excited state. Before deriving the correction factor for this type of transition, it is necessary to define the continuum frequency - dependant absorption coefficient (κ_ν) as

$$\kappa_\nu(p) = N(p) h\nu B_\nu \left[1 - \frac{N(p)_e}{N(p)} \exp(-h\nu/kT_e) \right] \quad (68)$$

where the second term in the bracketed expression of equation (68) can, if necessary, be approximated as small compared to one. The source function is

$$S_\nu = \frac{N(p)_e}{N(p)} \frac{2h\nu^3}{c^2} \left[\exp(h\nu/kT_e) - \frac{N(p)_e}{N(p)} \right] \quad (69)$$

and, the equation of radiative transfer is

$$\frac{dI_\nu}{ds} = \kappa_\nu (S_\nu - I_\nu) \quad (70)$$

where the subscript e denotes the number of atoms in the excited state p, if p were in equilibrium with the ions and electrons at the local T_e and pressure. When in chemical and thermodynamic equilibrium

$$\frac{N(p)_e}{N(p)} = 1 \quad (71)$$

and the source function becomes the black body function (B_ν). For this type of transition, since excited states are in equilibrium, $N(p) = N(p)_e$. Then, by the law of mass action

$$N(p)_e = \frac{N_e N_I g_p h^3}{2(2\pi m_e kT_e)^{3/2} Q_{eI}} \exp(x_p/kT_e) \quad (72)$$

By the definition of the degree of ionization and the degree of dissociation, equation (72) can be re-written as

$$N(p)_e = \frac{4N_\omega^2 \beta^2 \alpha^2}{2 \left[\frac{2\pi m_e kT_e}{h^2} \right]^{3/2} \frac{\exp(-x_p/kT_e)}{g_p} Q_{eI}} \quad (73)$$

Now, by definition

$$\frac{N_A}{2N_\omega \beta (1-\alpha)} = 1 \quad (74)$$

By multiplying the right - hand side of equation (73) by the left - hand side of equation (74) and making the assumption that the number density of the ground state is equal to the number density of atoms (N_A), the law of mass

action yields

$$N(p) = N(p)_e = \frac{\beta \alpha^2}{(1-\alpha)} \frac{(1-\alpha_e)}{\beta_e \alpha_e^2} N_A \left[\frac{Q_A}{g_p} \exp(\epsilon_x/kT_e) \right]^{-1} \quad (75)$$

and the absorption coefficient becomes

$$\kappa_\nu = \frac{\beta \alpha^2}{(1-\alpha)} \frac{(1-\alpha_e)}{\beta_e \alpha_e^2} N_A \left[\frac{Q_A}{g_p} \exp(\epsilon_x/kT_e) \right]^{-1} h\nu B_\nu \times [1 - \exp(-h\nu/kT_e)] \quad (76)$$

By redefining the terms after N_A on the right - hand side of equation (76) as the radiative cross-section (σ), the absorption coefficient can be written as

$$\kappa_\nu = \frac{\beta \alpha^2}{(1-\alpha)} \frac{(1-\alpha_e)}{\beta_e \alpha_e^2} N_A \sigma \quad (77)$$

and the source function becomes the black body function.

For a transition from a continuum state to the ground state, $N(1)$ does not equal $N(1)_e$. However, the assumption that $N(1) = N_A$ still holds. The absorption coefficient, equation (68), then becomes

$$\kappa_\nu = N_A \sigma \quad (78)$$

where the cross - section is defined as before.

Next, utilizing the law of mass action once again, the equilibrium number density of the ground state can be found from

$$N(1)_e = \frac{N_e N_I g_p h^3}{2 \left(2\pi m_e k T_e \right)^{3/2} Q_{eI}} \exp(\epsilon / k T_e) \quad (79)$$

By definition

$$N_e = N_I = 2N_\omega \beta \alpha \quad (80)$$

and,

$$\frac{N(1)_e}{N(1)} = \frac{4N_\omega^2 \beta^2 \alpha^2 g h^3 \exp(\epsilon / k T_e)}{2 \left(2\pi m_e k T_e \right)^{3/2} Q_{eI}} \frac{N_A}{2N_\omega \beta (1-\alpha)} \quad (81)$$

From the Saha equation, it can be shown that

$$\frac{2N_\omega \beta \alpha_e^2}{(1-\alpha)} = 2 \left[\frac{2\pi m_e k T_e}{h^2} \right]^{3/2} \frac{Q_{eI}}{Q_A} \exp(-\epsilon / k T_e) \quad (82)$$

Substituting equation (82) into (81) and reducing the resulting expression yields

$$\frac{N(1)_e}{N(1)} = \frac{\beta \alpha^2}{(1-\alpha)} \frac{(1-\alpha_e)}{\beta \alpha_e^2} \quad (83)$$

The substitution of equation (83) into the definition of the source function, equation (69), results in

$$S_\nu = \frac{\beta \alpha^2}{(1-\alpha)} \frac{(1-\alpha_e)}{\beta \alpha_e^2} B_\nu \quad (84)$$

For line radiation, the absorption coefficient and source function are defined as

$$K_\nu = N(p) B_{pq} B_\nu h\nu \left[1 - \frac{N(q)}{N(p)} \exp(-h\nu / k T_e) \right] \quad (85)$$

$$S_\nu = \frac{N(q)}{N(p)} \frac{g_p}{g_q} \frac{2h\nu^3}{c^2} \left[1 - \frac{N(q)}{N(p)} \exp(-h\nu/kT_e) \right] \quad (86)$$

Since both excited states are assumed to be in equilibrium at the local electron temperature the source function can be replaced by the black body function. The absorption coefficient is the same as equation (77).

In the models of radiative heat transfer, the atomic terms in the first four bands of the Olstad model, the first band of the KCN and Anderson models, and selected equations of the radiance model involve transitions to the ground state, or a state very close to the ground state, and their black body functions were modified accordingly. In the remaining bands, the atomic terms in the Olstad model and the remaining bands in the other models all involved transitions to an excited state. Their absorption coefficients were similarly corrected for the LTNE condition. The values of α_e and β_e were found from a method suggested by Fay and Kemp¹⁷ as functions of T_e , while α and β were determined by adding the appropriate number densities found from the chemical reactions. The actual degree of ionization is simply the ratio of the number density of ions to the sum of the number densities of ions, atoms, and twice the number density of the molecules. The actual degree of dissociation is the ratio of the sum of the number densities of the ions and atoms to the sum of the ions, atoms, and twice the number density of the molecules.

Molecular Correction Factors

A LTNE condition similar to that previously

outlined for atomic radiative processes will exist if it is assumed that the electronically excited molecules are in equilibrium with their atomic counterparts at the local T_h and pressure. The later essentially assumes that dissociation is dominated by collisional phenomena.

If a transition from an upper excited state, B, to a lower excited state, A, is hypothesized, the absorption coefficient can be expressed as

$$\kappa_\nu = N(B)\sigma \left[1 - \exp\left[-h\nu/kT_e\right] \right] \quad (87)$$

where the bracketed term on the right-hand side of equation (87) is approximately one, and

$$N(B) = N_{N_2}(B) \quad (88)$$

The problem is how to express the number density of the excited diatomic nitrogen [$N_{N_2}(B)$] in terms of the number density of the ground state diatomic nitrogen [$N_{N_2}(1)$] that is obtained in the modified Grose code.

First, assuming that the population of the B state is in equilibrium with the atoms and that the number density of the ground state is approximately equal to the number density of diatomic nitrogen, then, from the law of mass action,

$$\frac{N_N^2}{N(B)} = \frac{Q_N^2}{Q_B} \exp\left[-\epsilon_{NB}/kT_h\right] \quad (89)$$

or,

$$N(B) = \frac{N_N^2 Q_B}{Q_N^2} \exp\left[\epsilon_{NB}/kT_h\right] \quad (90)$$

Now, in equilibrium

355

$$\frac{1-\beta_e}{4N_e\beta_e^2(1-\alpha_e)^2} = \frac{Q_{N_2}}{Q_N^2} \exp\left[\epsilon_{1N}/kT_h\right] \quad (91)$$

then, using the definitions of the degree of dissociation and the degree of ionization, and combining the above expressions, equation (90) becomes

$$N(B) = \frac{\beta^2(1-\alpha)^2}{(1-\beta)} \frac{(1-\beta_e)}{\beta_e^2(1-\alpha_e)^2} N_{N_2(1)} \left[\frac{Q_P}{Q_{N_2}} \exp(-\epsilon_{1B}/kT_e) \right] \quad (92)$$

where the bracketed term on the right-hand side of equation (92) is defined as the radiative cross-section. The absorption coefficient then becomes

$$\kappa_\nu = \frac{\beta^2(1-\alpha)^2}{(1-\beta)} \frac{(1-\beta_e)}{\beta_e^2(1-\alpha_e)^2} N_{N_2(1)} \sigma_{eBA} \quad (93)$$

while the source function is simply the black body function.

The absorption coefficients for the molecular terms in the Olstad model and the radiance model were then modified to correct for the LTNE by the above method. The KCN and Anderson models had no molecular terms and were, therefore, left unchanged. The equilibrium and actual degree of ionization and degree of dissociation were found by the same method as the atomic correction factor values. However, the α_e 's and β_e 's were assumed to be functions of T_h . The actual degrees of ionization and dissociation were found by the method outlined in the previous section. It should be noted that the computer code was modified so that the atomic and molecular corrections could be used independently of each other, or not at all.

RESULTS AND DISCUSSION

The National Aeronautics and Space Administration plans to test the concept of aerobraking before beginning the construction of an AOTV. Consequently, NASA has proposed the use of an aeroassisted flight experiment (AFE) to test the reliability of the heat shield and the validity of the aerobraking concept. The AFE currently being studied by NASA is a 60° blunted cone that is approximately two meters in diameter. The original shock shape used in the Grose⁷ computer code was modified to yield an axisymmetric body that approximated that of the AFE. A ten species, eleven reaction chemistry model (Table 2) was incorporated

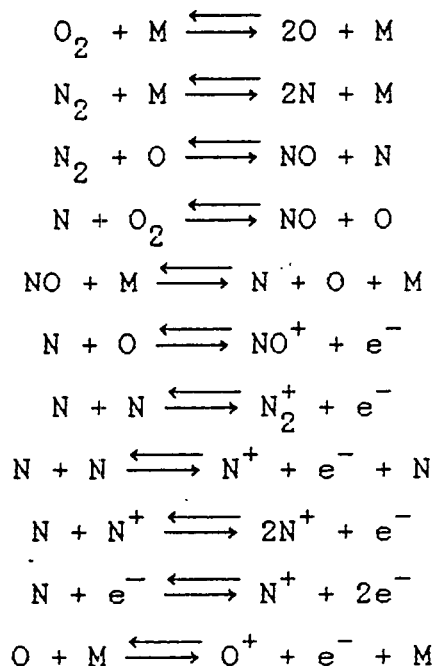


Table 2. Chemical Reaction Model

into the Grose code by Bobskill⁹. All of the following results were obtained from the modified Grose code for a velocity of ten kilometers per second at an altitude of 80 kilometers.

Shock Jump Chemistry Models

The addition of the two shock jump chemistry models described in this thesis gave a total of three shock jump conditions in the modified Grose code. The first condition, incorporated by Grose, assumed that no chemical reactions take place over the shock front. The second condition assumed that diatomic oxygen dissociates to a state of equilibrium with the local temperature at the shock front, while diatomic nitrogen remains frozen. The final shock jump condition assumed that both diatomic oxygen and diatomic nitrogen were in equilibrium with the local shock front temperature. The third shock jump condition was developed in this thesis, but not used since it was felt that the lower speeds of the AFE's flight regime would be out of the range of the third model's region of applicability.

In order to keep the computational effort reasonable, the flowfield solution method was only applied out to a maximum distance of 10 centimeters along the shock. and the flowfield was solved along individual streamlines separated by a distance of 0.5 centimeters. Solutions were obtained using both the first shock jump condition for a control, and the second to study differences in chemistry and the number of computational steps needed to reach $X = 10$ cm.

As Table 3 indicates, the use of the oxygen

Streamline	Computational Steps		Percentage Difference
	Condition 1	2	
1	808	895	10.77
2	718	702	- 2.23
3	514	508	- 1.17
4	393	373	- 5.09
5	350	334	- 4.57
6	294	327	11.22
7	234	214	- 8.55
8	206	190	- 7.77
9	191	175	- 8.38
10	177	165	- 6.78
11	150	153	2.00
12	115	121	5.22
13	105	84	-20.00
14	96	74	-22.92
15	84	67	-20.24
16	74	58	-21.62
17	66	50	-24.24
18	57	42	-26.32
19	41	35	-14.63

Table 3. Comparison of Computational Steps Taken
by Flowfield Solution Method to Reach
X = 10 cm.

equilibrium model yielded, for most streamline, substantial savings in the number of computational steps required to solve the flowfield. Some streamlines, however, had an increase in the number of steps needed for solution by the oxygen equilibrium model. Figures 2 and 3 compare the step

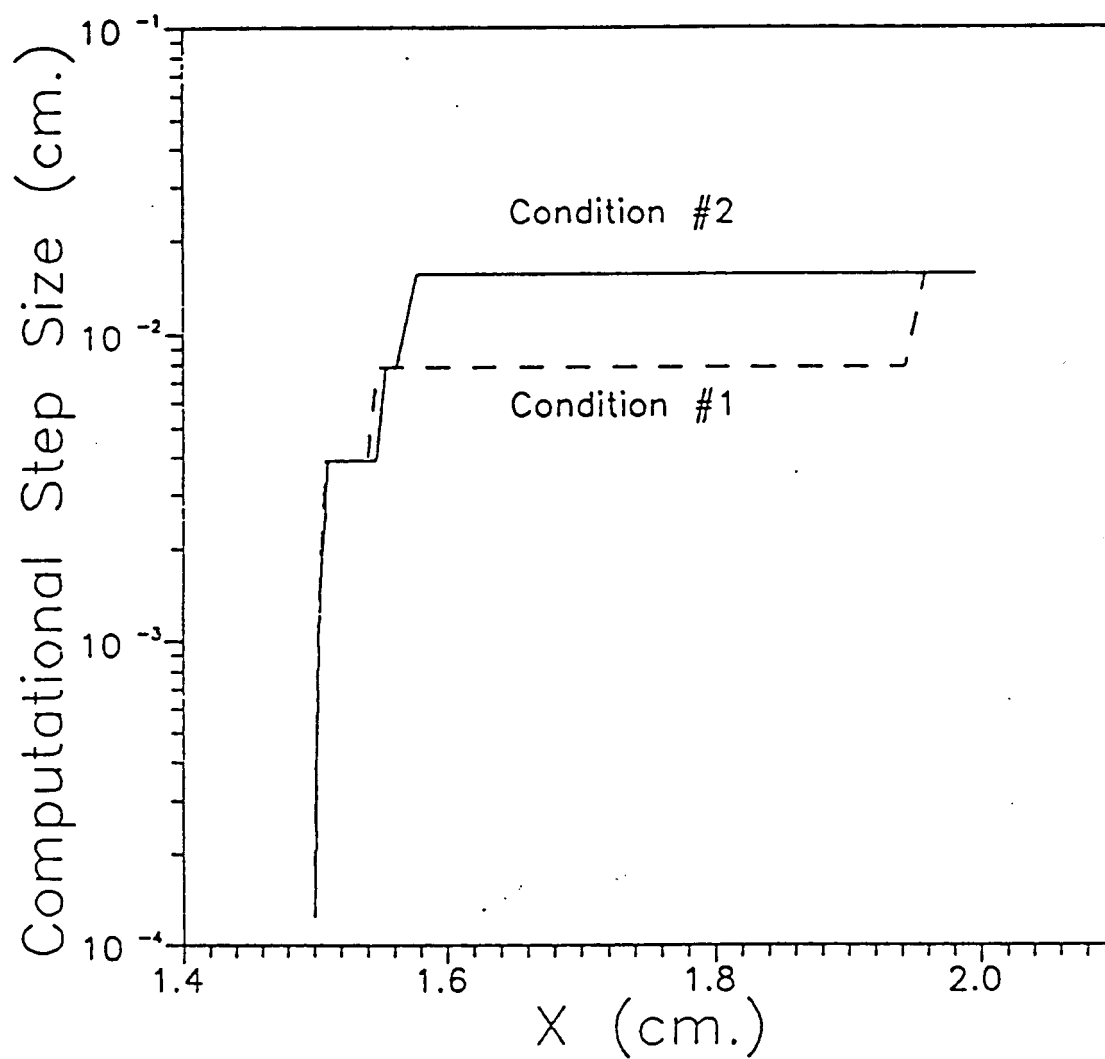


Figure 2. Computational Step Sizes Along Streamline 4.

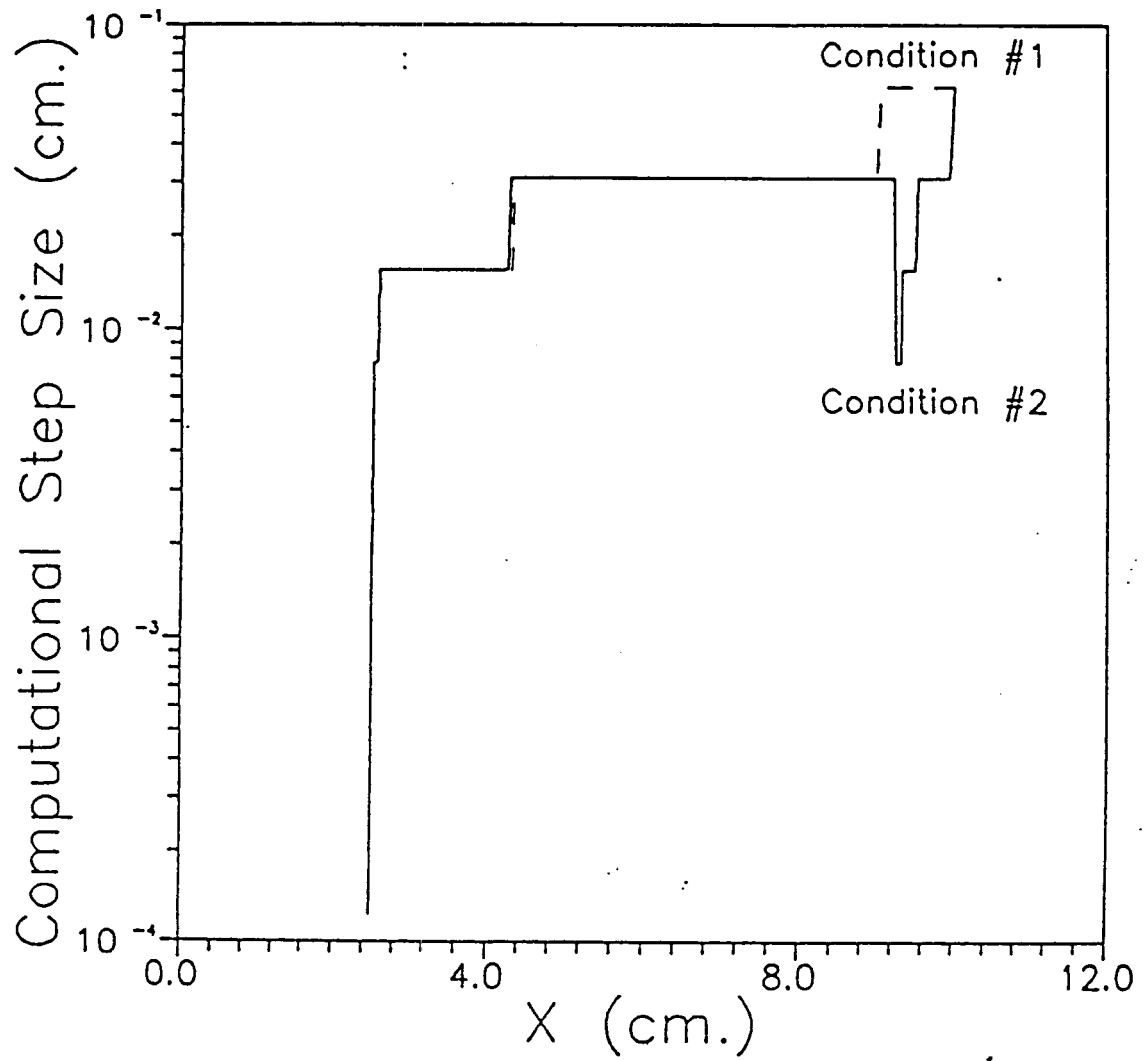


Figure 3. Computational Step Sizes Along Streamline 6.

sizes taken by the solution method along streamlines 4 and 6 (entering the shock front at $x = 1.5$ cm. and $x = 2.5$ respectively). Streamline 4 had a 5% reduction in the number of computational steps needed to solve for the flow properties along the streamline. Since the streamlines extend to an x of 10 cm., it can be seen that the savings realized by using the oxygen equilibrium model are in the nonequilibrium region immediately behind the shock front.

In contrast, Figure 3 indicates that the losses in the number of computational steps taken to calculate the flow properties along streamline 6 occur further downstream as the flow is approaching equilibrium. The step size increases along the streamline provided the chemical reactions do not cause the enthalpy to exceed the initial value by a given tolerance internal to the Grose code. It appears that the oxygen equilibrium model causes a minor imbalance in the enthalpy of some streamlines that requires the code to use smaller step sizes in order to maintain constant enthalpy. This effect could be the result of the tolerance values used in the original Grose code; but, unfortunately, it was not possible to investigate what effect a change in the tolerance values would have.

In Figure 4, it can be seen that the heavy particle temperature of the flow is initially lower at the shock front by using the oxygen equilibrium model. The lower temperatures are the result of the dissociation process "soaking up" much of the flow's energy. Downstream of the immediate shock front, the temperature closely follows the temperature of the frozen shock front chemistry condition. Similarly, Figures 5 and 6 indicate an over-dissociation of oxygen at the shock front. The concentrations of atomic and diatomic oxygen quickly resume the frozen chemistry values as the flow reaches equilibrium. Since the reaction rate of oxygen

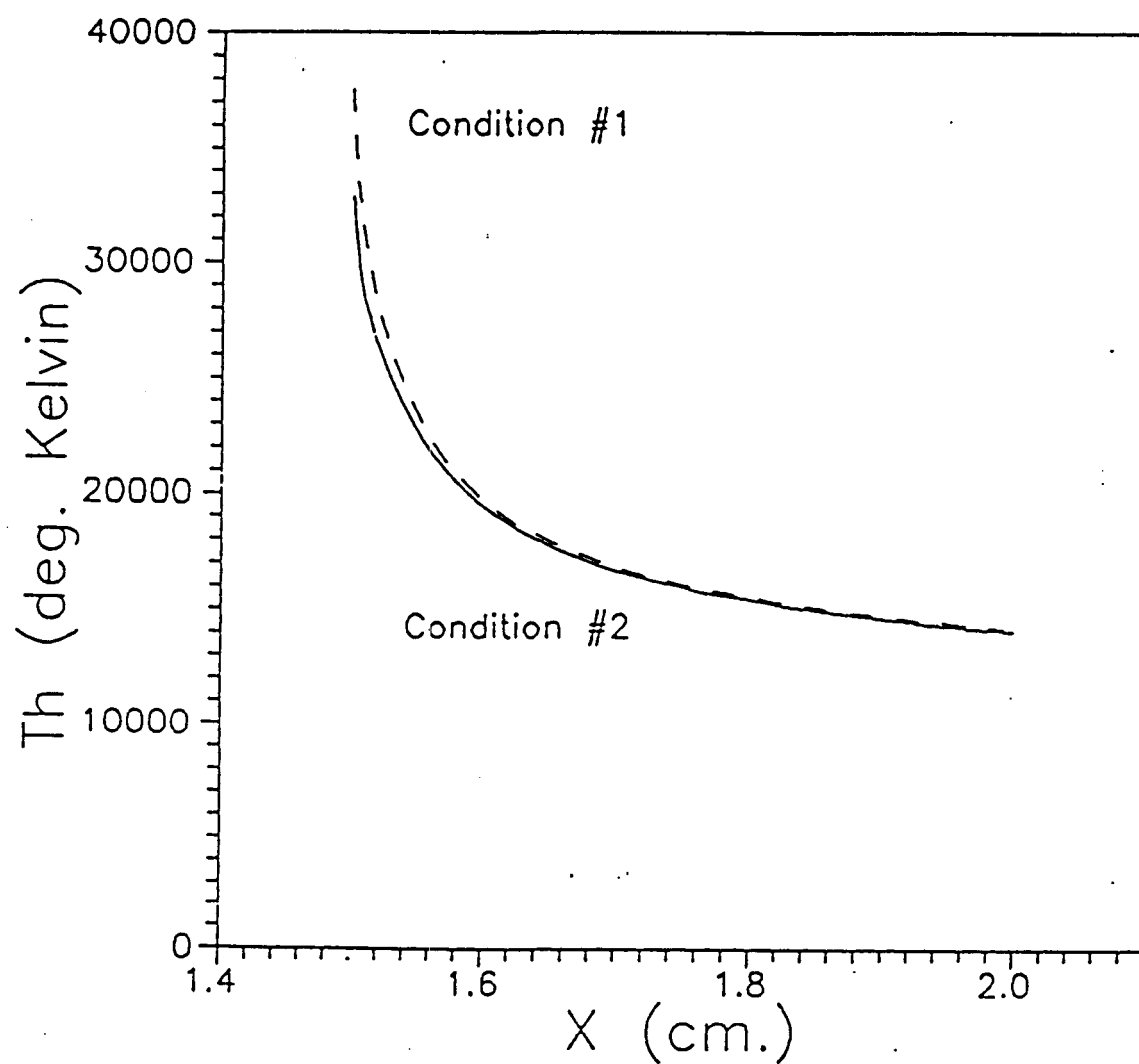


Figure 4. Comparison of Heavy Particle Temperatures for Shock Jump Conditions 1 and 2 on Streamline 4.

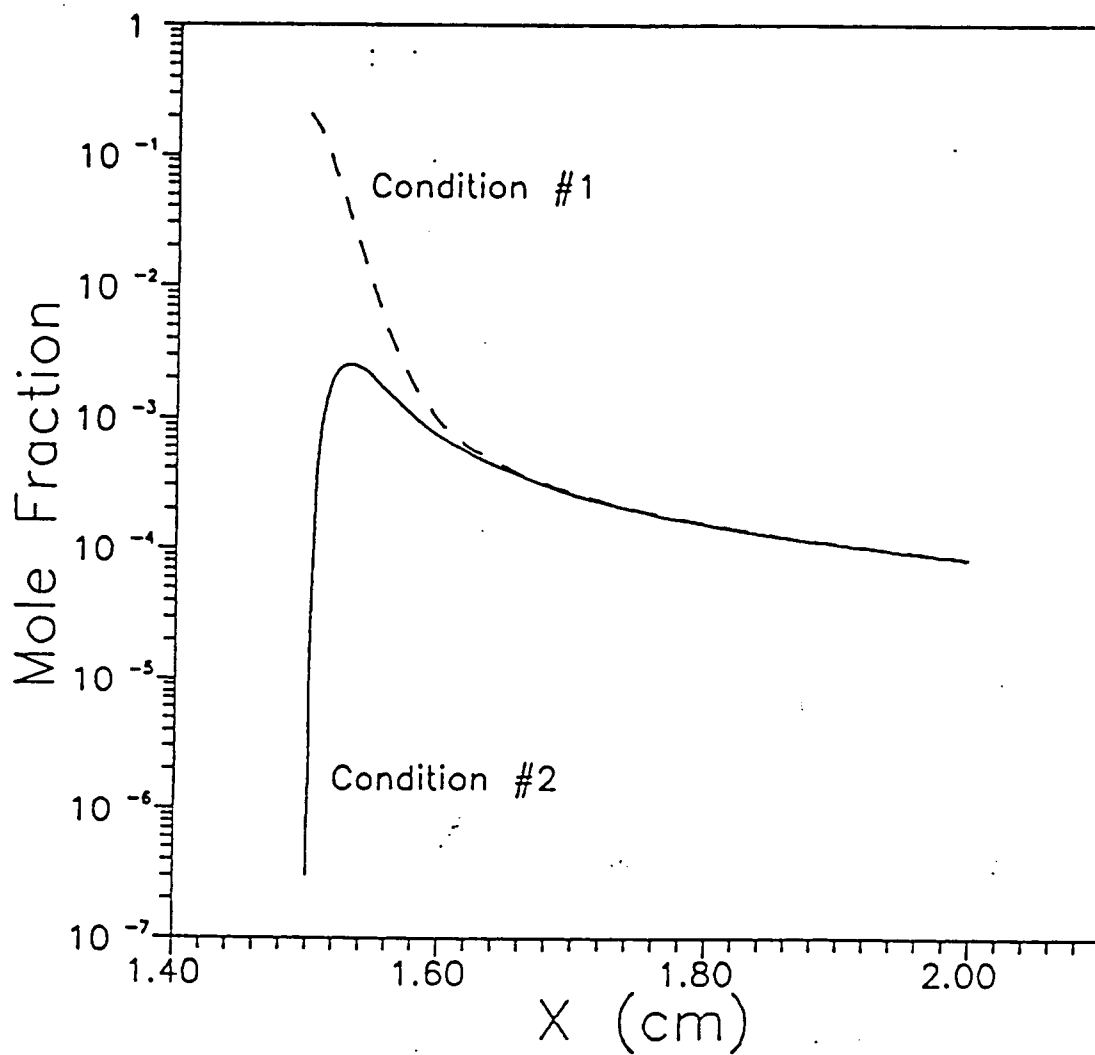


Figure 5. Comparison of O_2 Concentrations for Shock Jump Conditions 1 and 2 on Streamline 4.

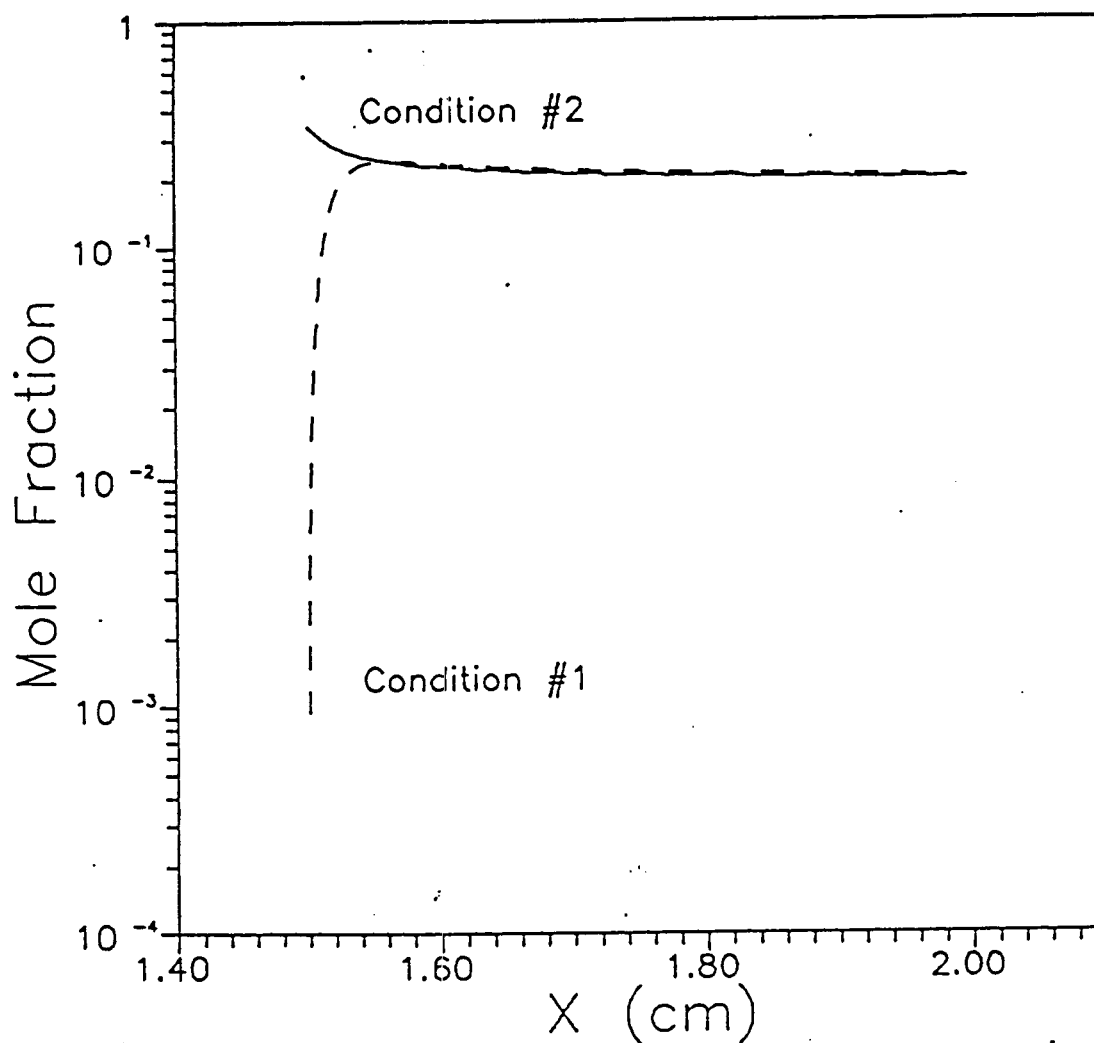


Figure 6. Comparison of Atomic Oxygen Concentrations for Shock Jump Conditions 1 and 2 on Streamline 4.

recombination is slower than the rate of oxygen dissociation, it is felt that the initial over-dissociation of oxygen allows the solution method to take larger computational steps in the nonequilibrium region. The larger computational step sizes would not be possible in the nonequilibrium region were it oxygen dissociation - dominated.

The most substantial savings in the number of steps taken to calculate the streamline properties were seen in the streamlines entering the shock layer further up the shock front where most of the flow is in chemical nonequilibrium. It was not expected that the oxygen equilibrium model would result in any savings of computational steps in the equilibrium region. Such savings were only expected in the nonequilibrium region, and the results support this expectation.

Electron Temperature Models

After the two electron temperature models were incorporated into the modified Grose code, several computer runs were made to compare the Carlson and Rieper² correlation with the Park³ assumption. In Figure 7, the Carlson and Rieper correlation is displayed for a freestream velocity of 10 km./sec at an altitude of 80 km. At this speed, the correlation yields the expected behavior for the T_e . The electron temperature has an initial sharp increase at the shock front followed by an almost linear pattern through the nonequilibrium region to the equilibrium region. However, the correlation fails to yield an electron temperature that matches the heavy particle temperature in the equilibrium region. It is also felt that the values of T_e are somewhat low for the AFE flowfield as compared to the results of others¹⁸. The original Carlson

and Rieper correlation was derived at a lower velocity than that encountered in the AFE's trajectory.

In order to give a fairer comparison, the correlation was also tested at a freestream velocity of 7.5 km./sec. and an altitude of 80 km. The results are plotted in Figure 8. Again, the correlation gives an electron temperature that conforms to the expected pattern, yet at the lower speed, the correlation's T_e matches the T_h in the equilibrium flow region. It is believed that the correlation breaks down at higher freestream velocities because the T_e is dependant on the number density of N_2 . At the AFE velocities, little N_2 is left in the flowfield, and as a consequence, the correlation breaks down in the 10 km./sec. case.

The Park assumption, where $T_e = T_{vN_2}$, was also tested at a freestream velocity of 10 km./sec. using the CVDV and CPL vibrational - dissociation coupling models. The results are plotted in Figures 9 and 10. The CVDV model gives an initially high electron temperature peak near the shock front followed by a decrease that approaches the T_h in the equilibrium region of the flow. The CPL model yields similar results, yet with a less severe T_e peak near the shock front. The CVDV model's high initial vibrational temperature for diatomic nitrogen is not believed to be reflective of the actual behavior of the T_e and could lead to excessive radiative heating from the region near the shock front in the case of a highly transparent shock layer, as shall be seen in the next section. It is therefore felt that the Park assumption combined with the CPL vibration - dissociation coupling model yields the best results for electron temperature of the two T_e models tested.

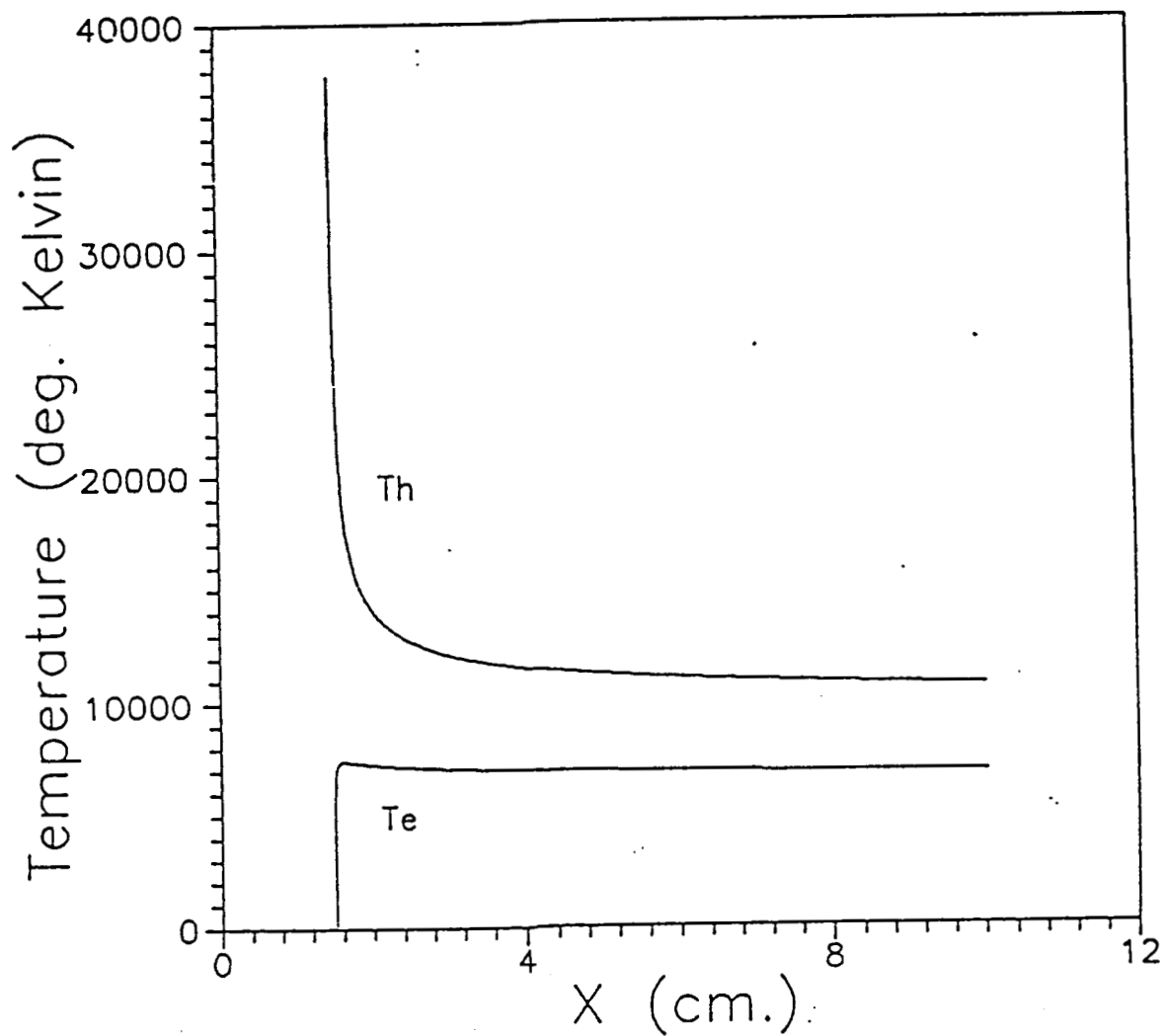


Figure 7. Carlson and Rieper T_e Correlation at 10 km./sec.
for Streamline 4.

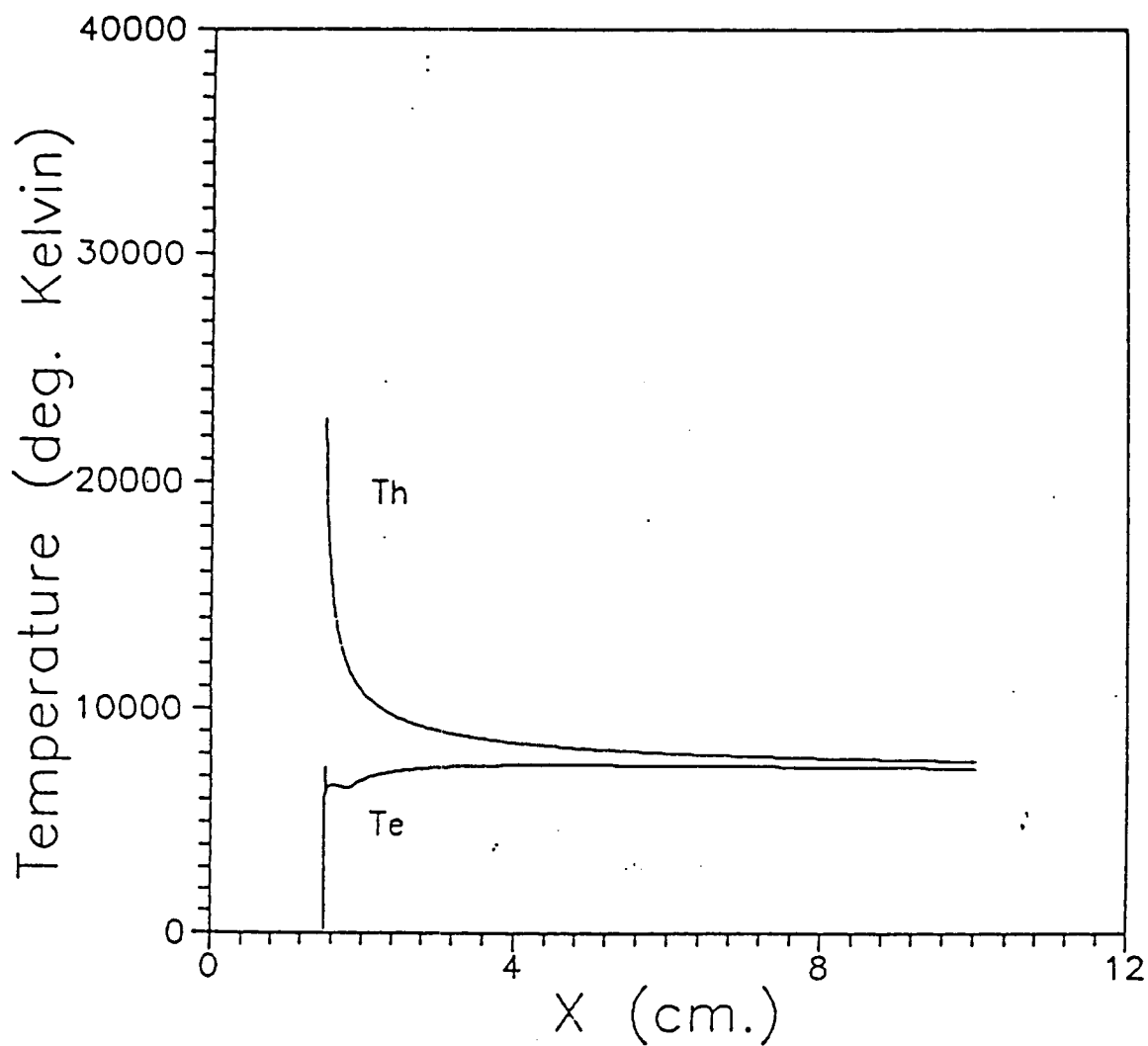


Figure 8. Carlson and Rieper T_e Correlation at 7.5 km./sec. for Streamline 4.

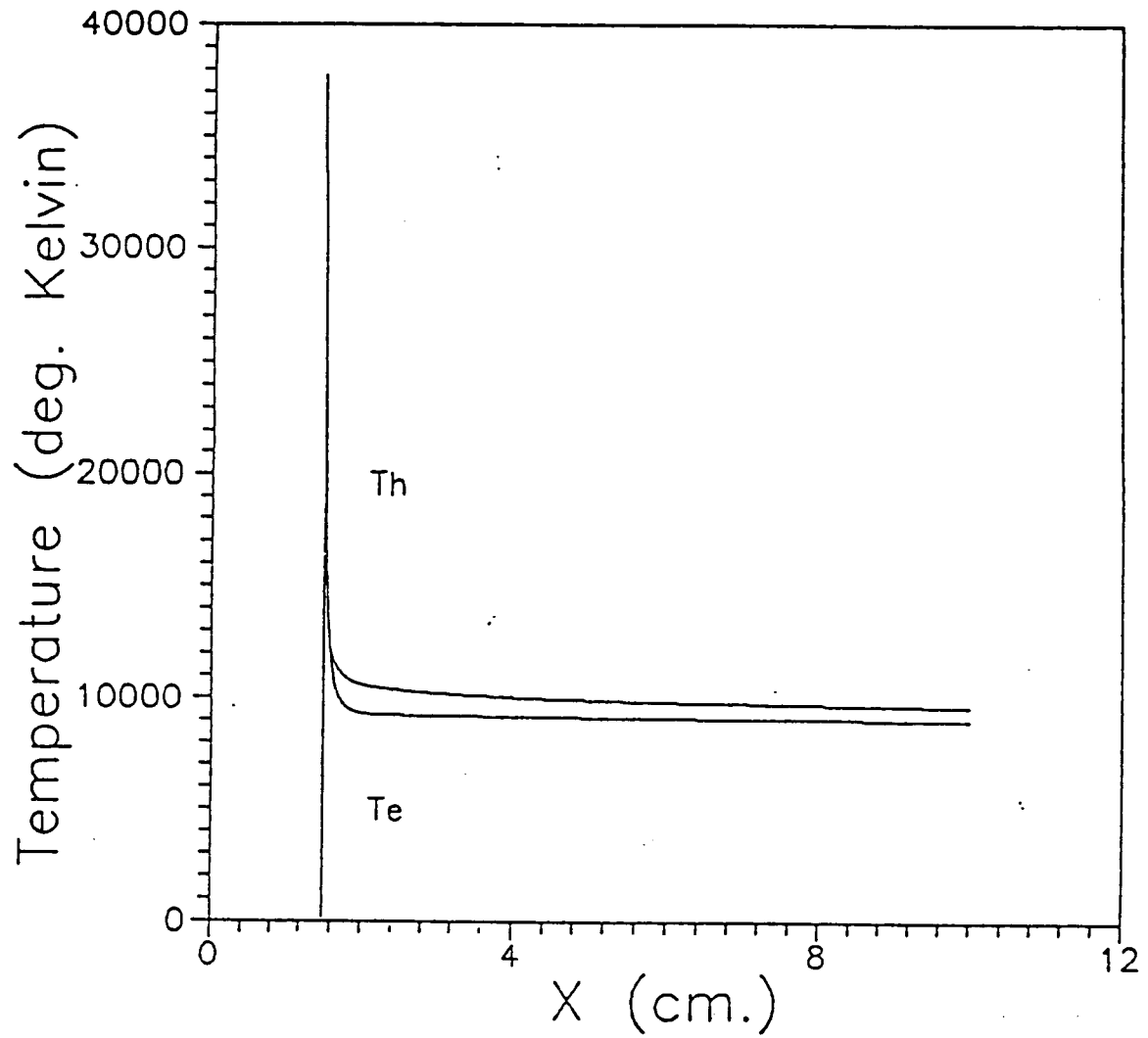


Figure 9. Park Assumption T_e at 10 km./sec. for Streamline 4 with CVDV Vibration-Dissociation Coupling Model.

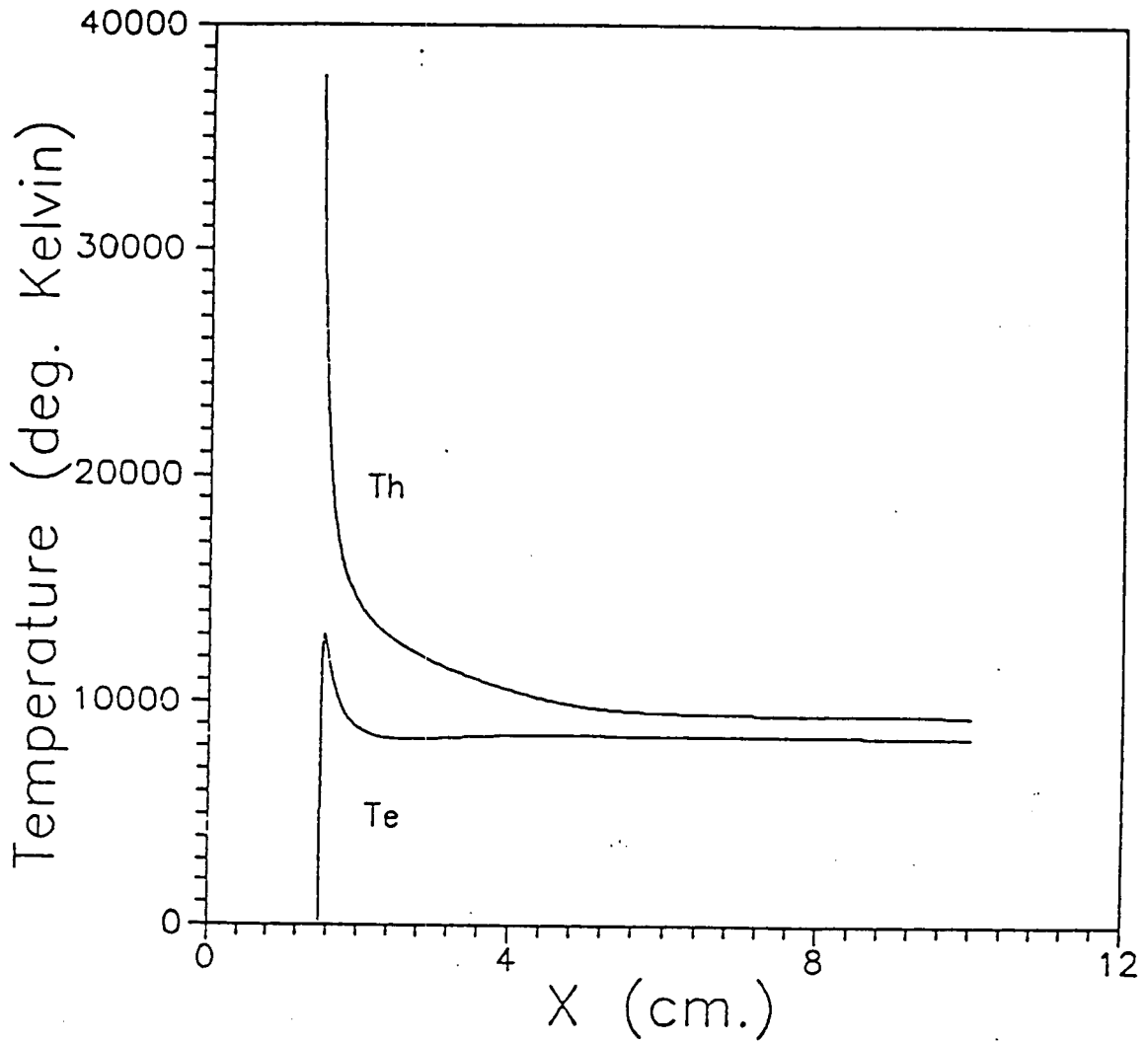
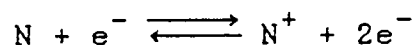


Figure 10. Park Assumption T_e at 10 km./sec. for Streamline 4 with CPL Vibration-Dissociation Coupling Model.

Radiative Heat Transfer Models and Nonequilibrium Correction Factors

The radiative heat transfer and the nonequilibrium correction factors are both highly dependent on the selected chemical reaction rates as well as the vibration - dissociation model. Several options were available with the modified Grose code⁹. In addition to the CVDV and CPL vibration - dissociation models, it was possible to choose for the electron impact reaction,



the chemical rate postulated by Kang and Dunn¹⁹, the Wilson rate²⁰ used by Carlson²¹, or the chemical rate proposed by the Sandia National Laboratory²². The Kang and Dunn rate is approximately an order of magnitude faster than the Wilson rate, and the Wilson rate is, in turn, an order of magnitude faster than the Sandia rate. Thus, there were several combinations of the above chemical rates, vibration - dissociation coupling models, and nonequilibrium correction factors available in the modified Grose code. All temperature, species number density, and correction factors were calculated and saved during the solution of the flowfield. The surface radiative heat transfer was then calculated along the body out to an x of 9.5 cm. For simplicity, results will only be presented at the location corresponding to 9.0 cm. along the shock front. In all cases, the electron temperatures used in this part of the study followed the Park assumption that $T_e = T_{vN_2}$. The electron temperatures and the number densities of N and N^+ are plotted against their distance from the shock front to the vehicle wall in Figures 11 - 19.

By inspection of Figures 11 - 13, it can be seen

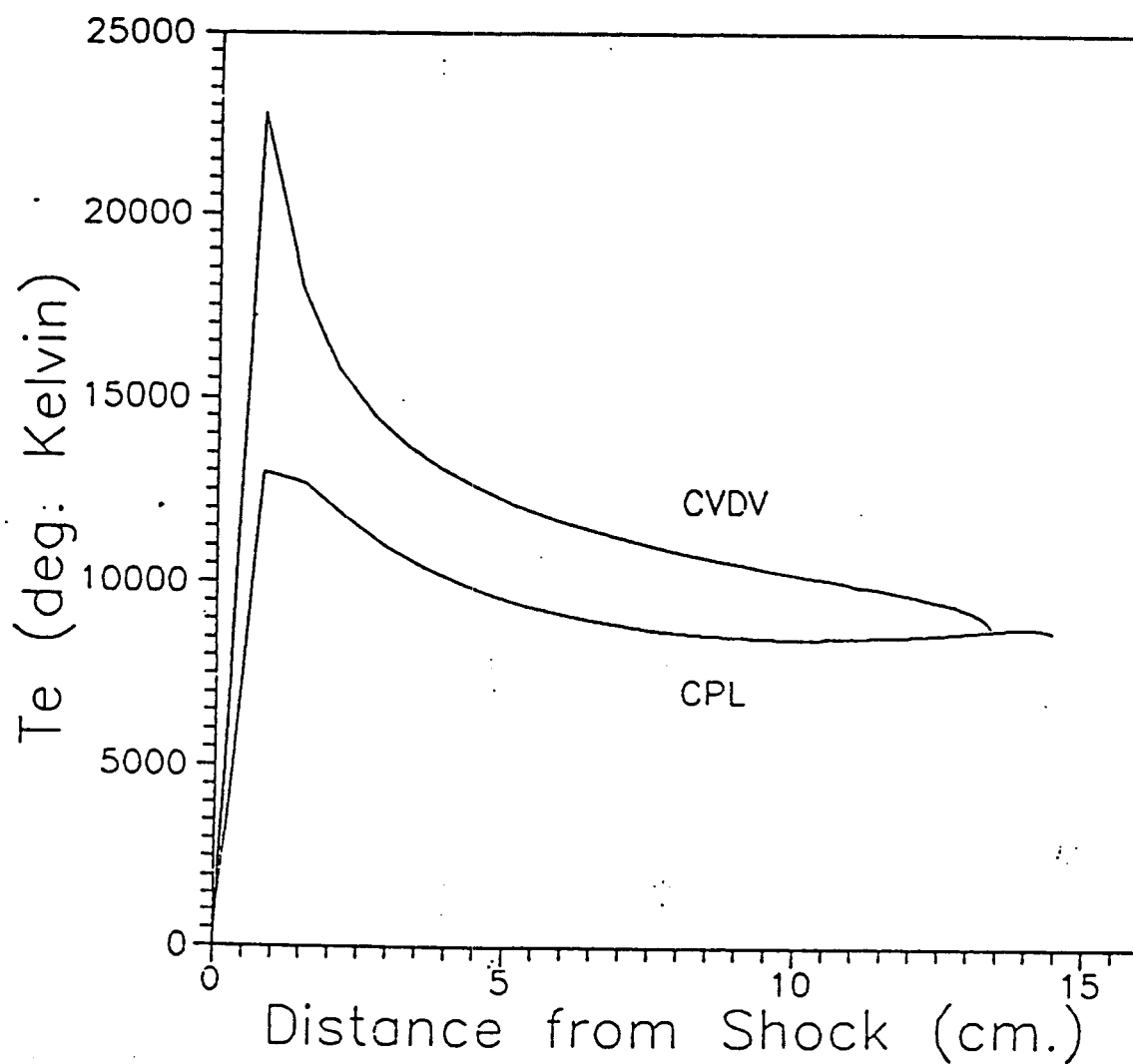


Figure 11. Comparison of T_e for Wilson Reaction Rate at $X=9.0$ cm.

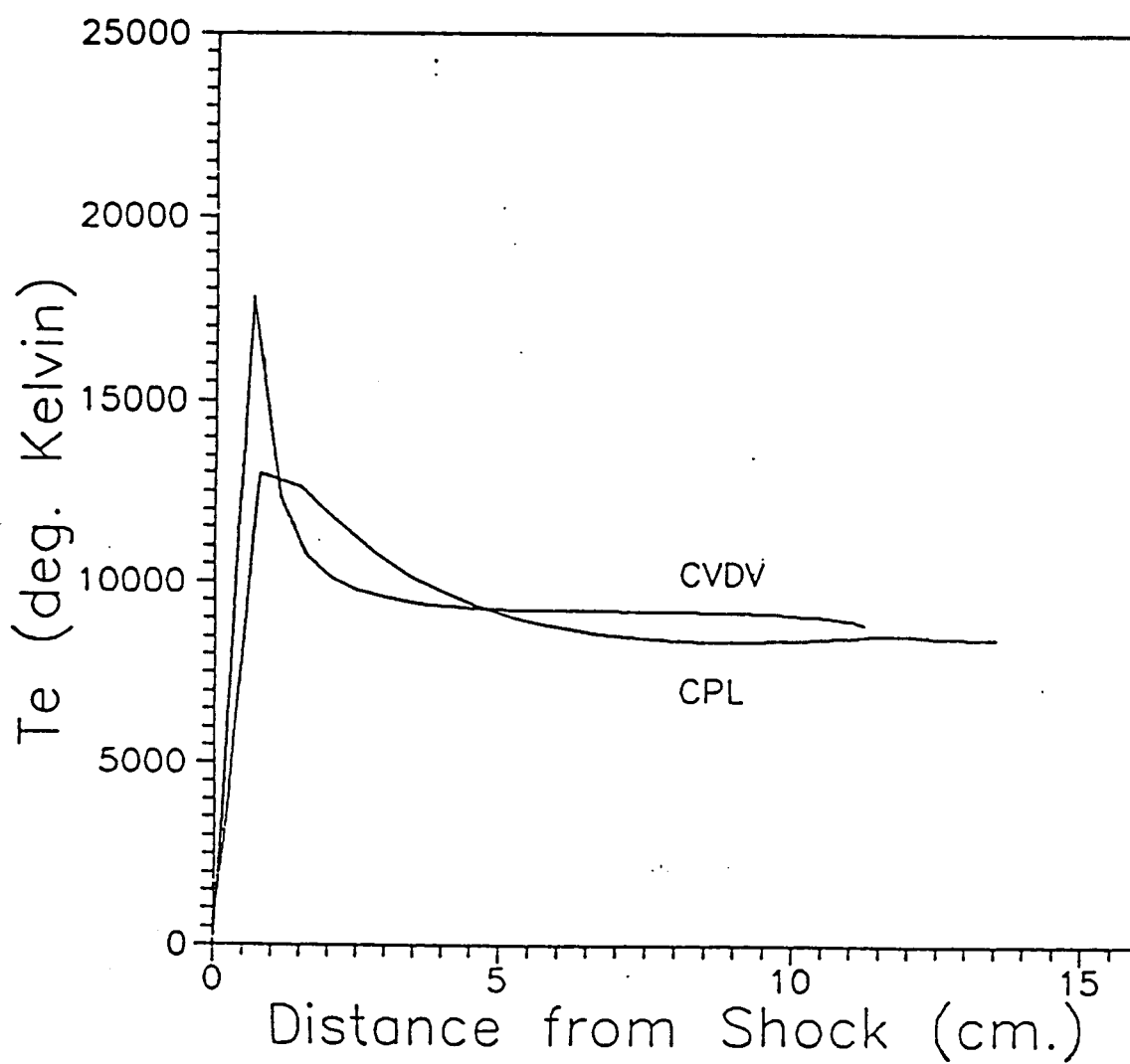


Figure 12: Comparison of T_e for Kang and Dunn Reaction Rate at $X=9.0$ cm.

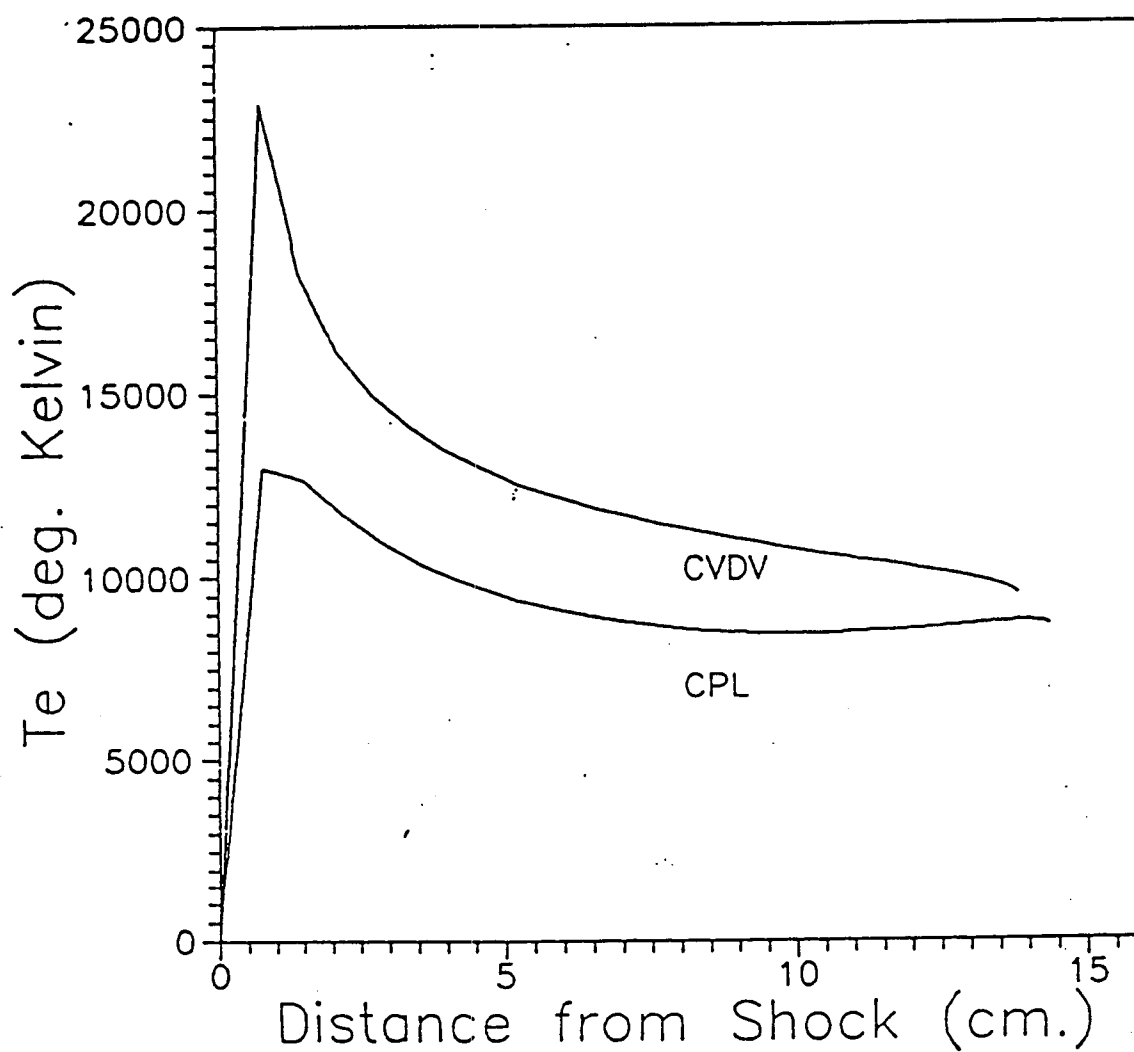


Figure 13. Comparison of T_e for Sandia Reaction Rate at $X=9.0$ cm.

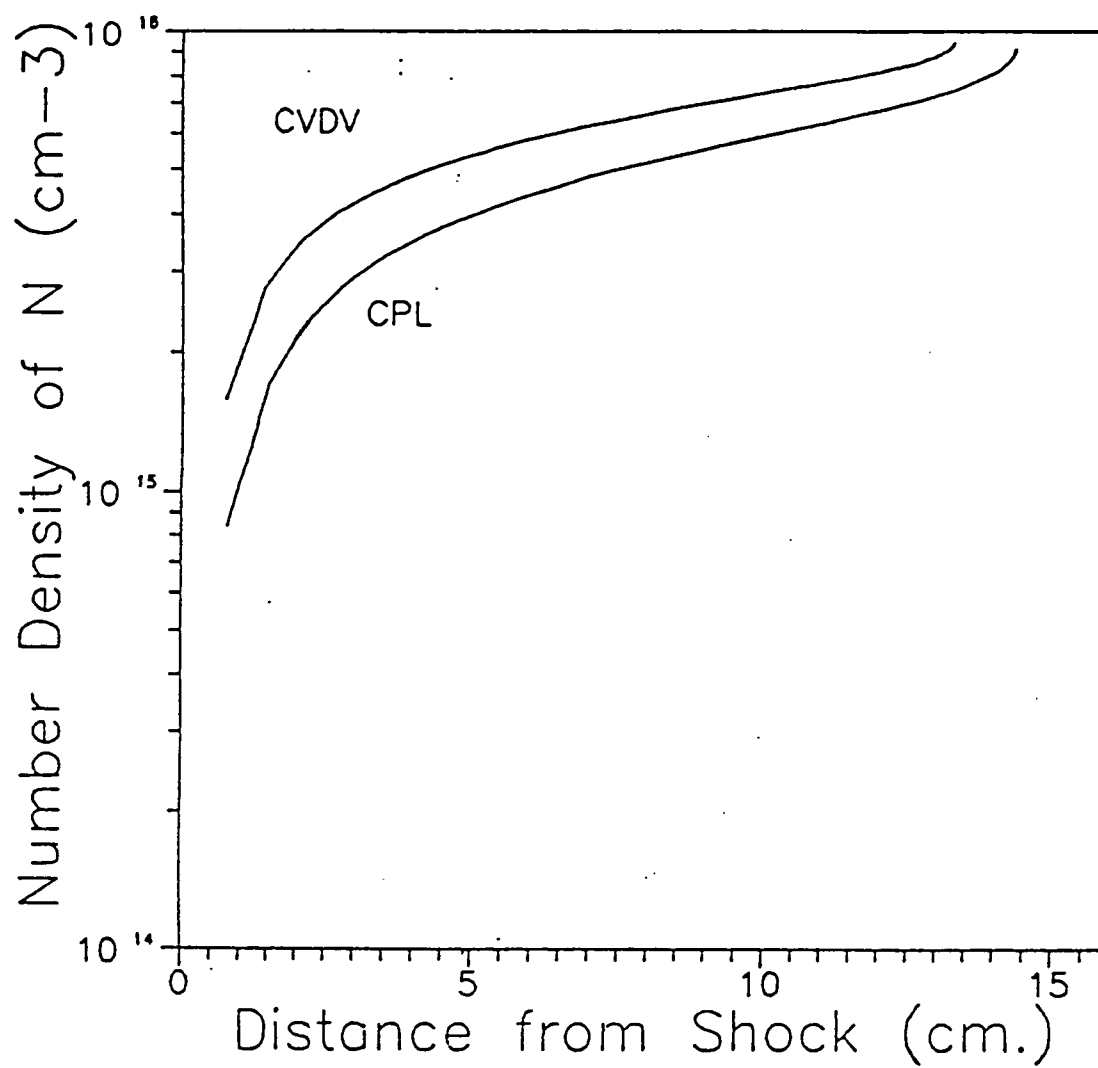


Figure 14. Comparison of N_N for Wilson Reaction Rate at $X=9.0$ cm.

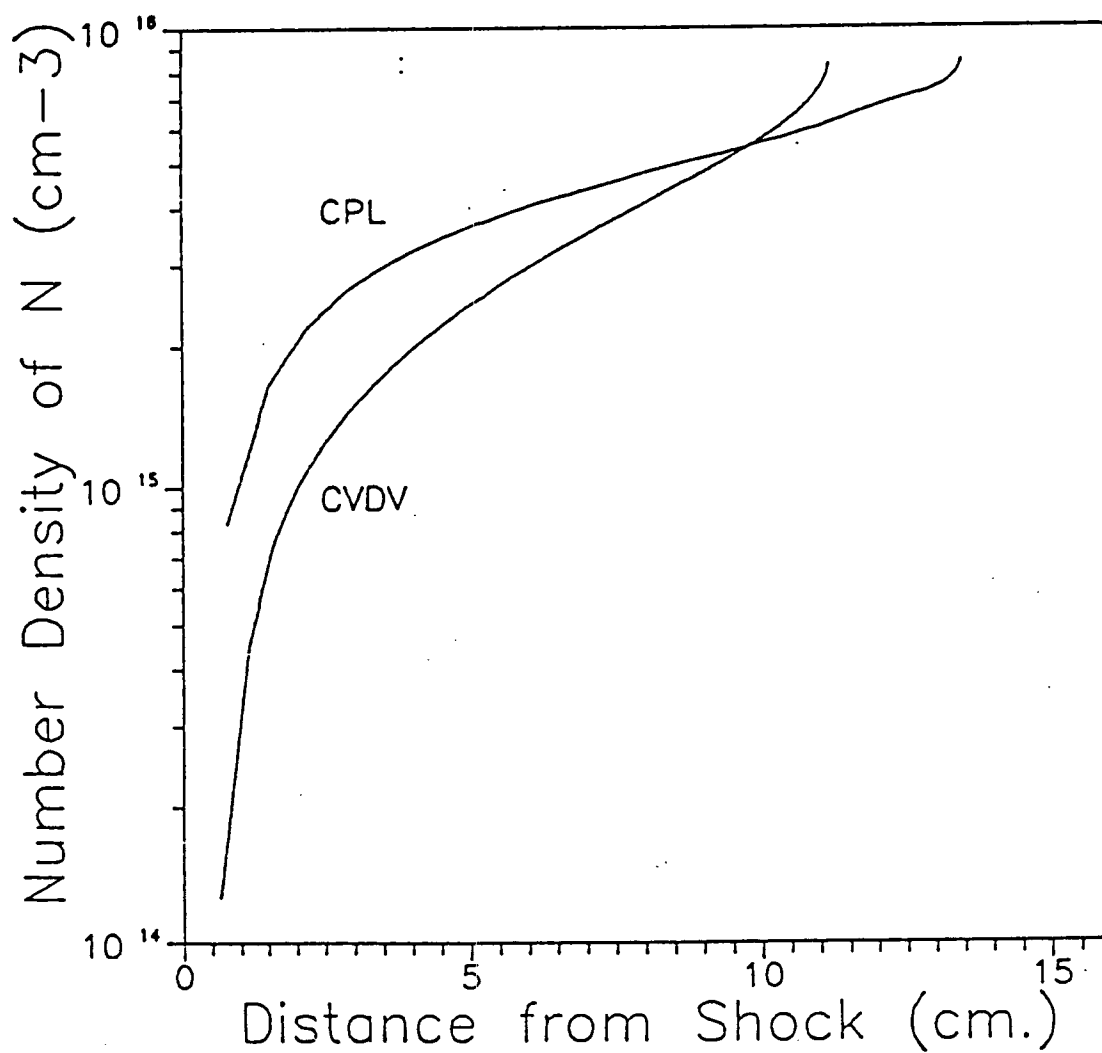


Figure 15. Comparison of N_N for Kang and Dunn Reaction Rate at $X=9.0$ cm.

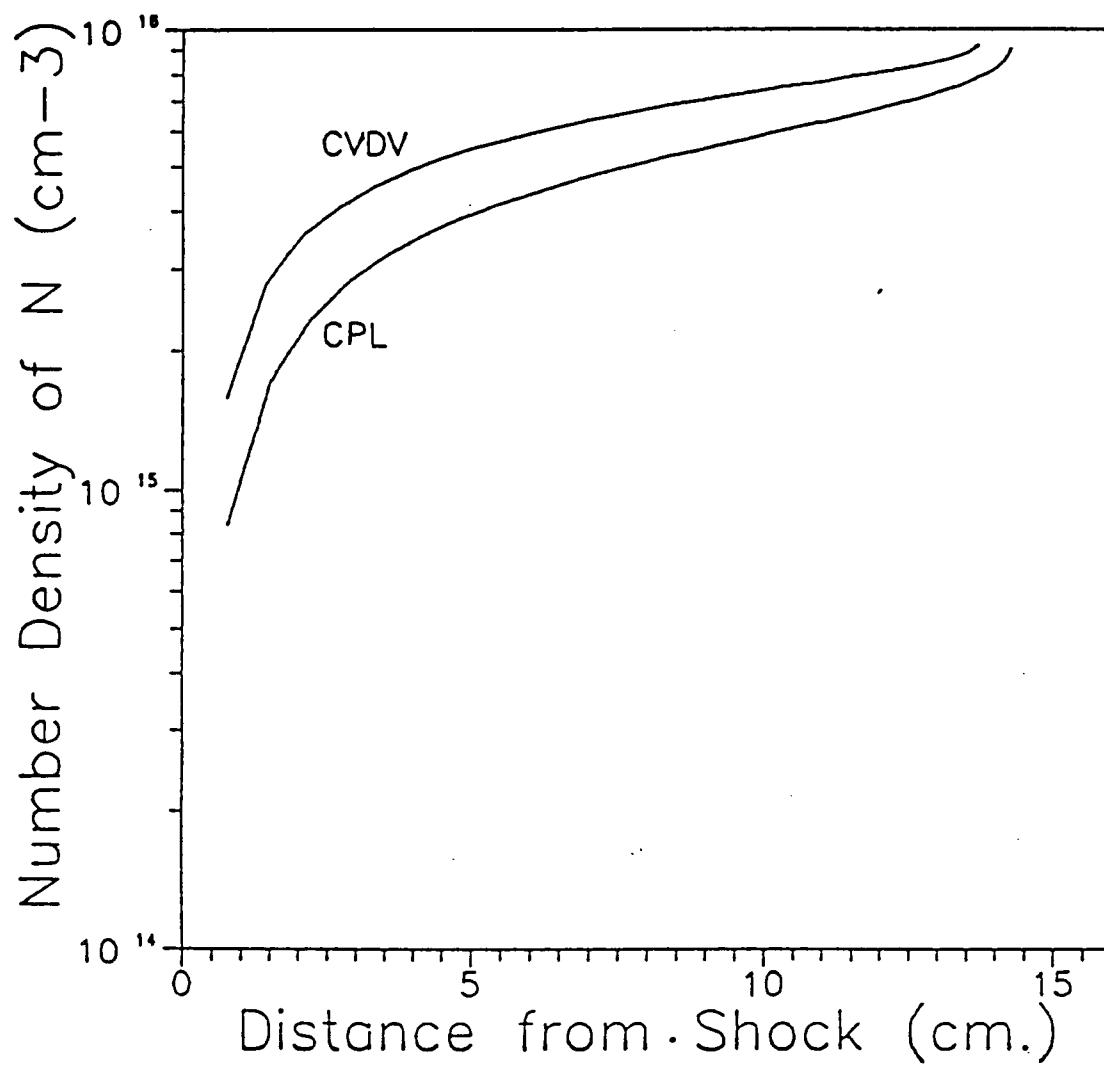


Figure 16. Comparison of N_N for Sandia Reaction Rate at $X=9.0$ cm.

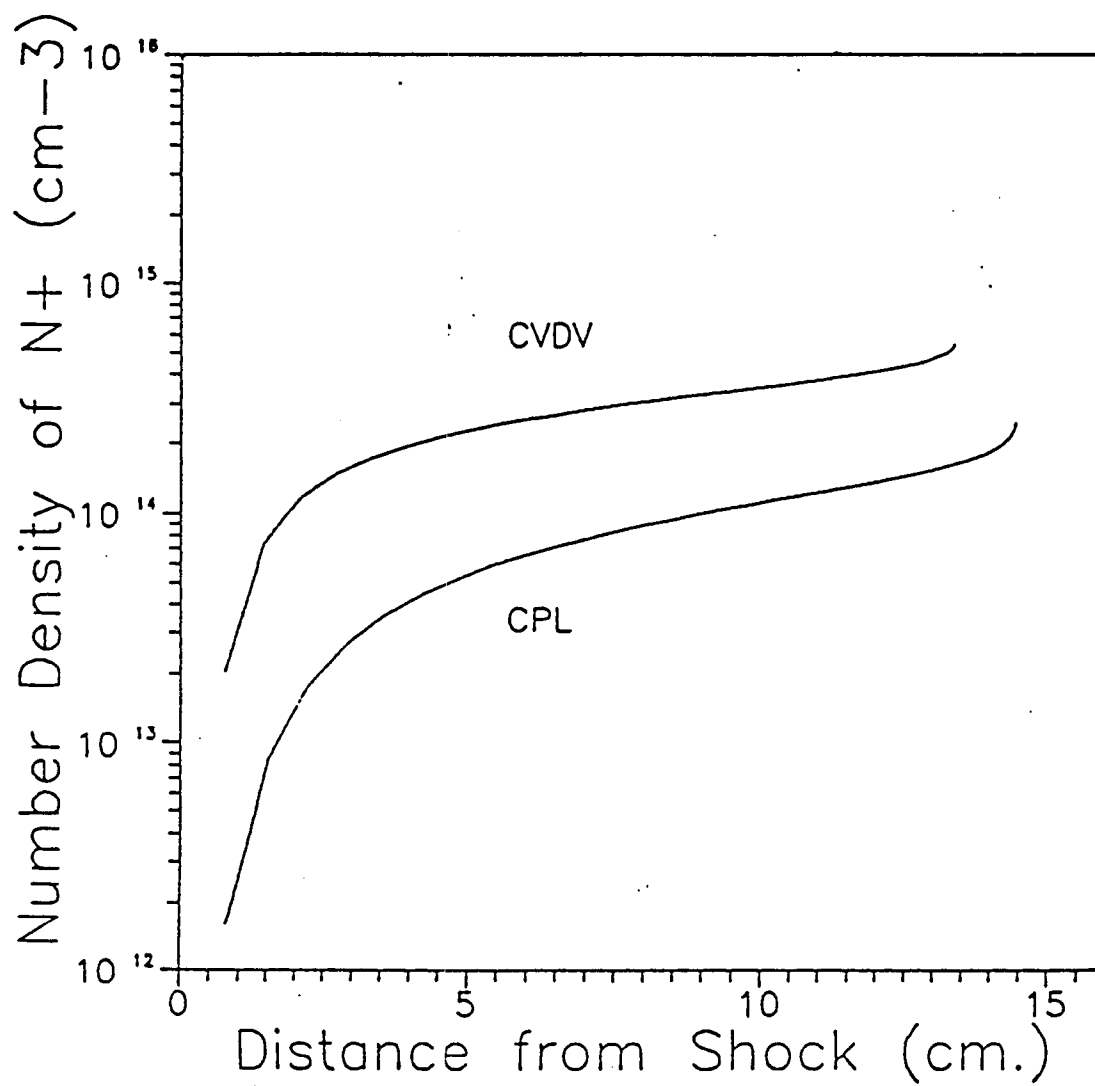


Figure 17. Comparison of N_{N^+} for Wilson Reaction Rate at $X=9.0$ cm.

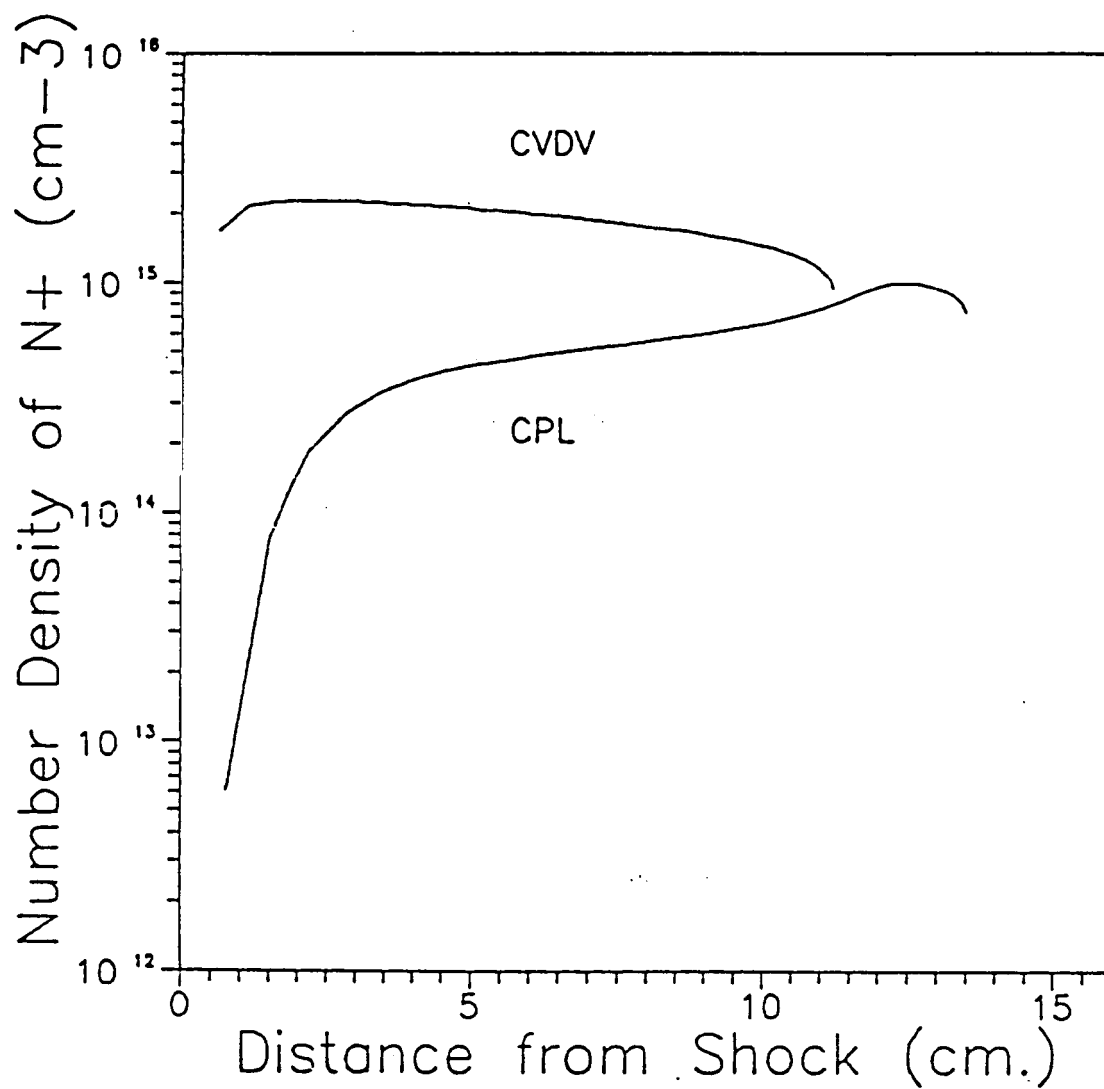


Figure 18. Comparison of N_{N^+} for Kang and Dunn Reaction Rate at $X=9.0$ cm.

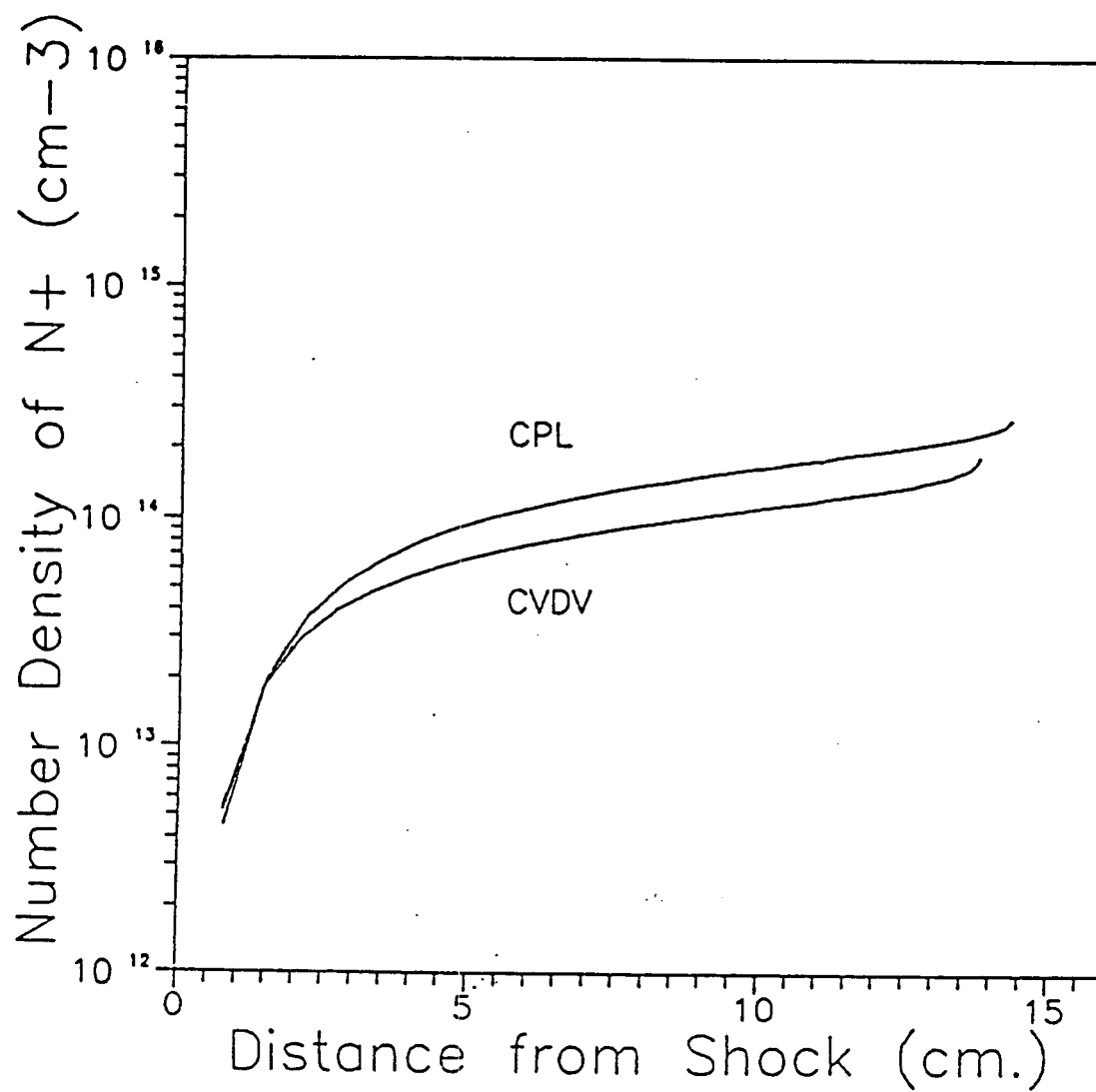


Figure 19. Comparison of N_{N+} for Sandia Reaction Rate at $X=9.0$ cm.

that the CPL model consistently gives lower electron temperatures than the CVDV models. As was stated earlier, it is felt that the actual behavior of the electron temperature is best modelled by the CPL model. The high T_e near the shock front in the CVDV model could lead to an over-estimate of the radiative heat transfer. On the other hand, the CPL model yields fairly consistent results for T_e regardless of the chemical rates chosen.

The number densities of atomic nitrogen for the Wilson and Sandia rates are fairly consistent with one another; and the CVDV model yields a larger concentration of N than the CPL model for both rates. However, the Kang and Dunn rate yields a higher initial rate of ionization of nitrogen relative to the other chemical rates. It should be noted that in the original formulation of the Wilson rate the ionization of atomic nitrogen was a two step process. The first step raised the atomic nitrogen to an excited state, while the second step involved rapid ionization. The first reaction step in this process is included in the chemistry model of the modified Grose code, while the second is not specifically accounted for. The Kang and Dunn reaction rate, on the other hand, represents the ionization of nitrogen in a one- step process; and thus, this process is accounted for in the modified Grose code's chemistry model. The combination of the Kang and Dunn rate with the CVDV vibration - dissociation coupling model predicts a very high concentration of N^+ and a lower concentration of N relative to the other chemical rates and coupling models. The number densities of N^+ predicted by the Wilson and Sandia rates are also fairly consistent with one another, except that the Wilson rate predicts a lower N^+ number density with the CPL model than are present at any other rate or coupling model.

The differences in the radiative heat transfer

predicted between the coupling models and chemical rates can be seen in Tables 4 - 7. The uncorrected radiative heat transfer models yield predictions that are clearly too large and conflict with those obtained in the accurate, yet more time consuming, Direct Simulation Monte Carlo (DSMC) method¹⁸, or in experimental shock tube results¹⁵. As a consequence, it is felt that the uncorrected radiative heat transfer models should be disregarded. The data obtained from the radiative heat transfer model with the atomic correction factors alone similarly predicts too high of a radiative flux to the wall and should also be viewed with some suspicion.

It is interesting to note that the early ionization of nitrogen predicted by the Kang and Dunn electron impact rate consistently yields a lower radiative flux than the Wilson rate when uncorrected. However, when corrected, Kang and Dunn leads to a higher radiative flux than the Wilson rate. Figures 20 - 23 display the correction factors for the nitrogen terms; and as can be observed, the correction factors for atomic nitrogen with the Kang and Dunn rate are often greater than one. This fact accounts for the higher radiative heat transfer with the Kang and Dunn ionization rate. The high degree of ionization from the Kang and Dunn rate leads to a corresponding increase in the number of atoms in the excited state, and consequently, radiative flux associated with the photo-deionization from the excited nitrogen. If the chemistry model included the second step of the ionization process used by Wilson, it is possible that the Wilson rate predictions would correspond to the results yielded by the Kang and Dunn rate. Since the Kang and Dunn rate inherently take the two steps of the ionization process into account, it is felt that the Kang and Dunn rate yields the most accurate results given the chemistry

Radiative Heat Transfer (Watts/cm.²)

<u>Model/Band</u>	<u>Uncorrected</u>	<u>Atomic Corrections</u>	<u>Atomic & Molecular Corrections</u>
Olstad			
1	9.8	8.6	0.09
2	1059.0	434.0	0.05
3	3691.0	838.0	0.15
4	894.0	25.7	0.06
5	2938.0	2715.0	0.02
6	8040.0	4703.0	0.35
7	0.03	0.03	0.03
8	1838.0	1415.0	0.16
Total	18470.0	10140.0	0.91
KCN			
1	10.7	5.44	5.44
2	241.5	0.06	0.06
3	3054.0	2.32	2.32
4	918.0	0.02	0.02
5	284.0	0.15	0.15
Total	4510.0	7.99	7.99
Radiance			
1	0.01	0.01	0.01
2	0.79	0.79	0.79
3	0.49	0.49	0.49
4	0.2E-4	0.2E-4	0.2E-4
5	0.3E-17	0.3E-17	0.3E-17
6	0.3E-17	0.3E-17	0.3E-17
7	349.9	0.02	0.02
8	2860.0	39.6	39.6
9	358.6	0.02	0.02
10	1269.0	8.9	8.9
11	0.1E-15	0.1E-15	0.1E-15
12	0.1E-15	0.1E-15	0.1E-15
13	23.0	23.0	0.4E-3
Total	4863.0	72.8	49.78
Anderson			
1	8022.0	0.38	0.38
2	104800.0	0.3E-2	0.3E-2
Total	112800.0	0.39	0.39

Table 4. Q_r for Wilson Rate and CVDV Coupling Model at
X=9.0 cm.

Radiative Heat Transfer (Watts/cm.²)

<u>Model/Band</u>	<u>Uncorrected</u>	<u>Atomic Corrections</u>	<u>Atomic & Molecular Corrections</u>
Olstad			
1	0.17	0.10	0.05
2	7.3	4.5	0.13
3	15.8	8.1	0.02
4	3.9	0.50	0.9E-2
5	84.7	83.2	0.03
6	43.6	24.7	0.55
7	0.02	0.02	0.02
8	67.0	61.0	0.36
Total	222.4	182.1	1.17
KCN			
1	6.5	1.21	1.21
2	2.1	0.01	0.01
3	90.4	0.01	0.01
4	0.9	0.2E-2	0.2E-2
5	4.5	0.03	0.03
Total	104.4	1.27	1.27
Radiance			
1	0.4E-2	0.4E-2	0.4E-2
2	0.14	0.14	0.14
3	0.11	0.11	0.11
4	0.3E-4	0.3E-4	0.3E-4
5	0.9E-18	0.9E-18	0.9E-18
6	0.1E-17	0.1E-17	0.1E-17
7	1.33	0.3E-2	0.3E-2
8	595.2	28.9	28.9
9	2.27	0.09	0.09
10	278.3	263.5	263.5
11	0.7E-16	0.7E-16	0.7E-16
12	0.1E-15	0.1E-15	0.1E-15
13	10.9	10.9	0.8E-4
Total	888.2	303.6	292.7
Anderson			
1	53.0	0.05	0.05
2	424.4	0.4E-4	0.4E-4
Total	477.5	0.05	0.05

Table 5. Q_r for Wilson Rate and CPL Coupling Model at
X=9.0 cm.

Radiative Heat Transfer (Watts/cm.²)

<u>Model/Band</u>	<u>Uncorrected</u>	<u>Atomic Corrections</u>	<u>Atomic & Molecular Corrections</u>
Olstad			
1	0.38	0.55	2.82
2	88.27	66.11	0.79
3	146.9	137.1	4.88
4	11.36	8.7	2.39
5	744.1	732.0	0.61
6	685.3	559.4	11.2
7	0.05	0.04	0.04
8	351.0	334.2	6.06
Total	2027.0	1838.0	29.23
KCN			
1	8.71	29.9	29.9
2	1.77	2.49	2.49
3	33.77	30.26	30.26
4	2.11	0.58	0.58
5	2.65	6.51	6.51
Total	49.01	69.75	69.75
Radiance			
1	0.02	0.02	0.02
2	0.50	0.50	0.50
3	0.44	0.44	0.44
4	0.7E-4	0.7E-4	0.7E-4
5	0.9E-18	0.9E-18	0.9E-18
6	0.1E-17	0.1E-17	0.1E-17
7	2.0	0.87	0.87
8	296.4	2760.0	2760.0
9	34.45	0.1E-2	0.1E-2
10	416.8	1.76	1.76
11	0.6E-15	0.6E-15	0.6E-15
12	0.3E-15	0.3E-15	0.3E-15
13	0.05	0.05	0.1E-4
Total	750.6	2764.0	2764.0
Anderson			
1	853.5	72.53	72.53
2	15880.0	259.6	259.6
Total	16740.0	332.1	332.1

Table 6. Q_r for Kang and Dunn Rate and CVDV Coupling Model
at X=9.0 cm.

Radiative Heat Transfer (Watts/cm.²)

Model/Band	Kang & Dunn		Sandia	
	Rate		Rate	
	CPL Model	CVDV Model	CPL Model	
Olstad.				
1	0.40	0.02	0.06	
2	0.18	0.03	0.13	
3	0.60	0.02	0.04	
4	0.28	0.07	0.02	
5	0.09	0.01	0.03	
6	2.09	0.11	0.58	
7	0.03	0.03	0.02	
8	1.45	0.04	0.38	
Total	5.13	0.26	1.26	
KCN				
1	20.75	0.46	1.64	
2	0.38	0.05	0.02	
3	0.65	0.31	0.04	
4	0.07	0.2E-2	0.4E-2	
5	1.20	0.01	0.07	
Total	23.05	0.80	1.78	
Radiance				
1	0.01	0.7E-2	0.5E-2	
2	0.30	0.28	0.15	
3	0.25	0.20	0.12	
4	0.6E-4	0.9E-5	0.3E-4	
5	0.8E-18	0.3E-17	0.9E-18	
6	0.1E-17	0.3E-17	0.1E-17	
7	0.10	0.2E-2	0.6E-2	
8	1187.0	2.25	59.34	
9	0.07	0.02	0.09	
10	258.7	5.77	257.4	
11	0.2E-15	0.7E-16	0.9E-16	
12	0.2E-15	0.8E-16	0.1E-15	
13	0.2E-4	0.1E-2	0.7E-4	
Total	1447.0	8.52	317.1	
Anderson				
1	2.23	0.04	0.11	
2	0.6E-3	0.2E-3	0.3E-3	
Total	2.23	0.04	0.11	

Table 7. Q_r with Atomic and Molecular Correction Factors
at $X=9.0$ cm.

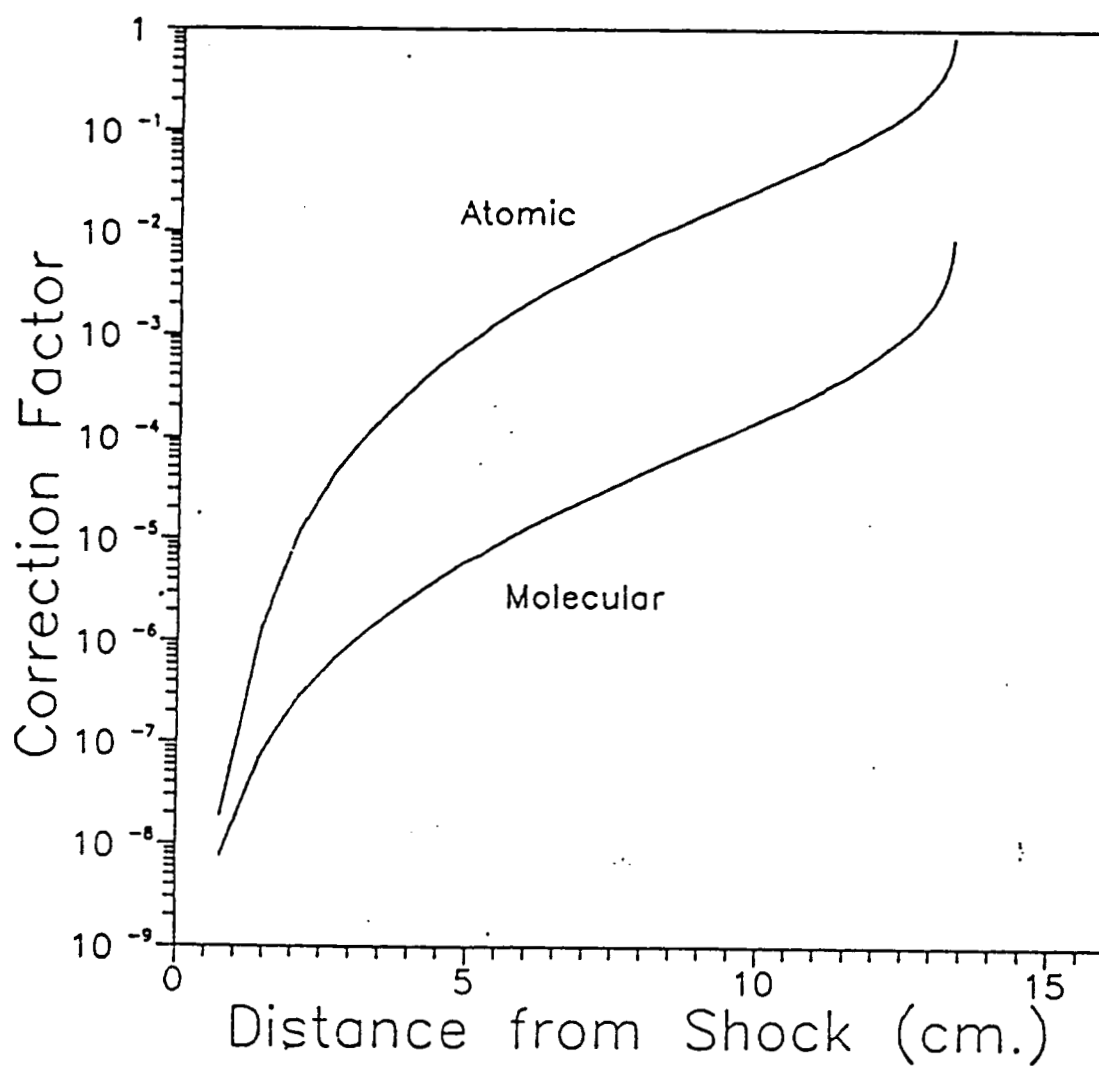


Figure 20. Nitrogen Correction Factors for Wilson Rate with CVDV Coupling.

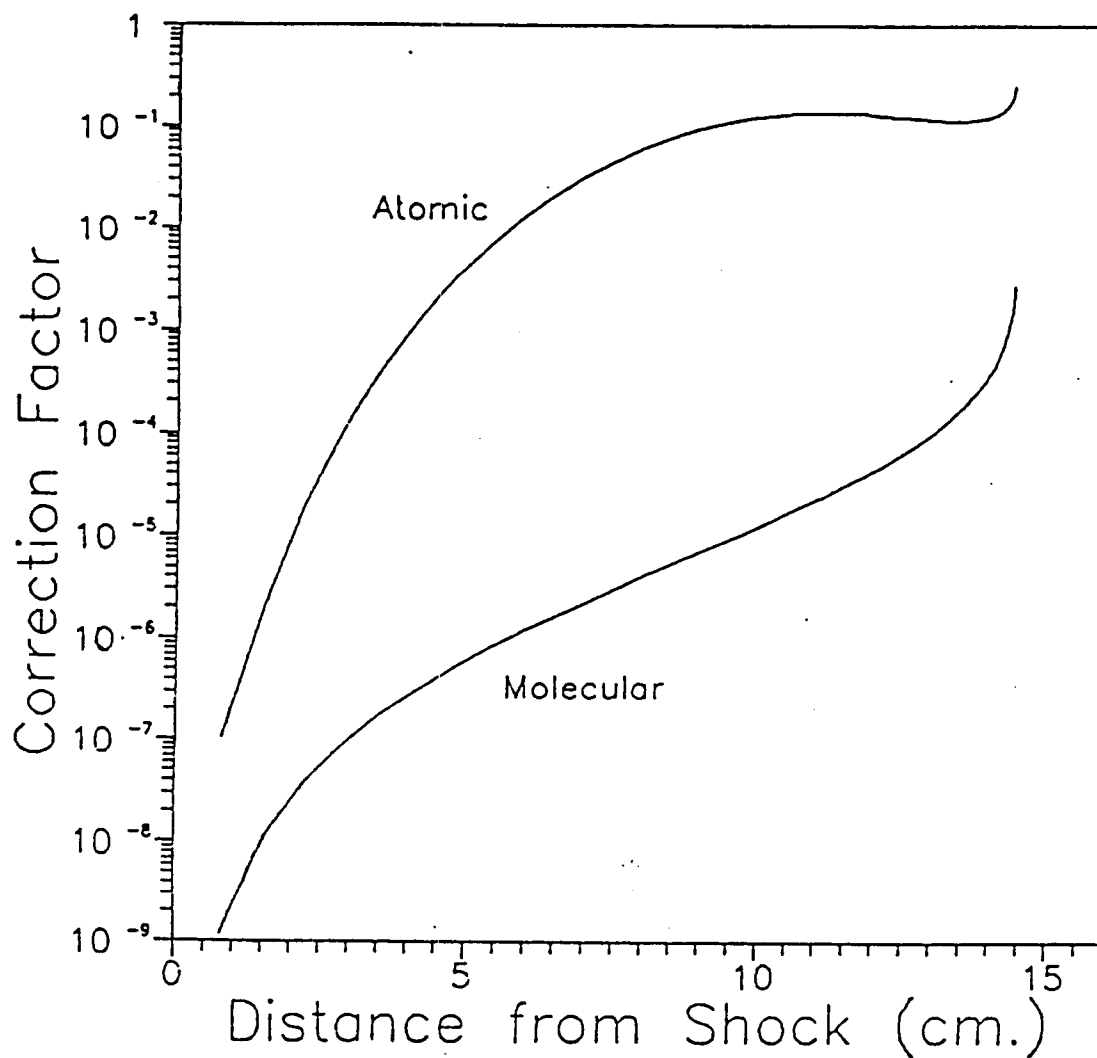


Figure 21. Nitrogen Correction Factors for Wilson Rate with CPL Coupling.

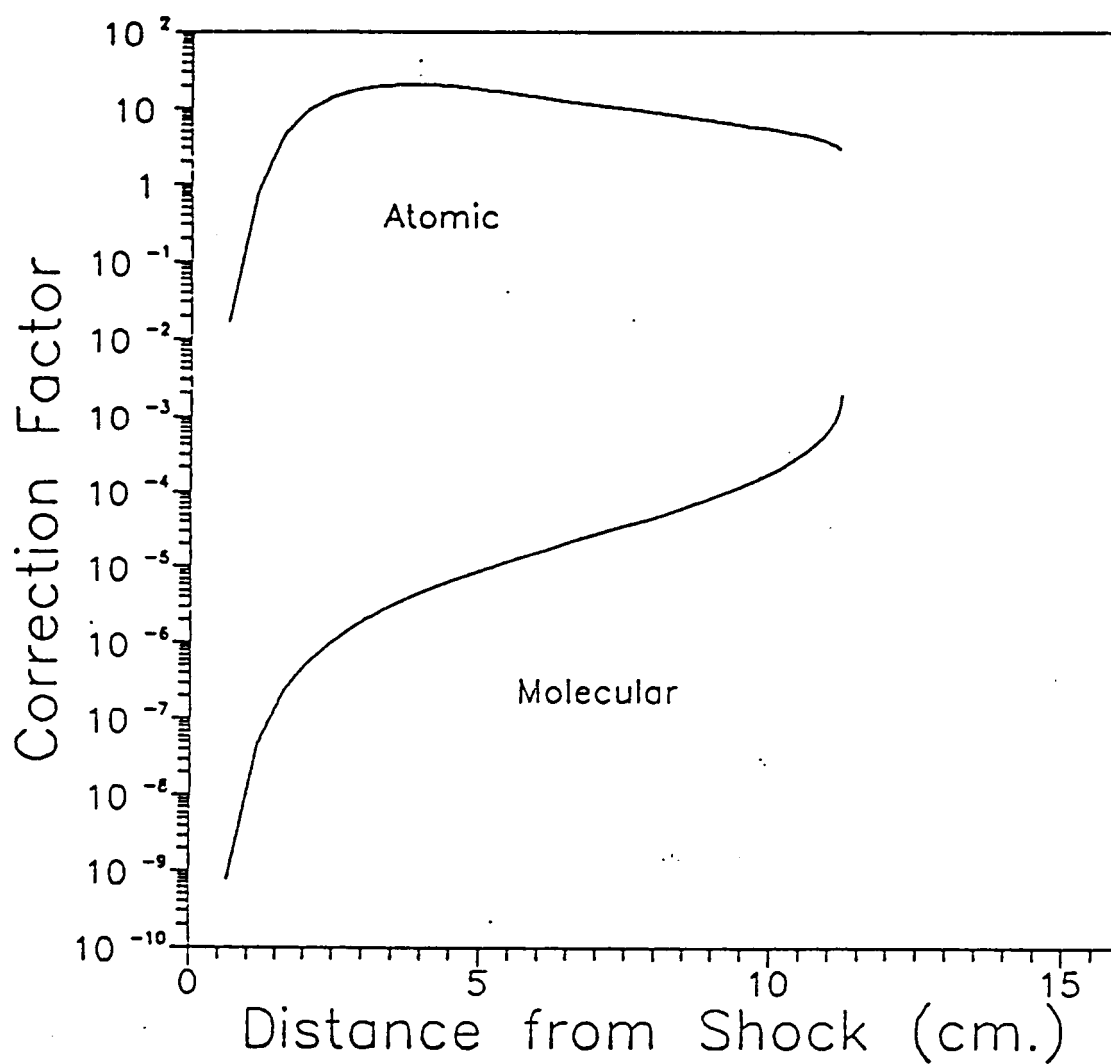


Figure 22. Nitrogen Correction Factors for Kang and Dunn Rate with CVDV Coupling.

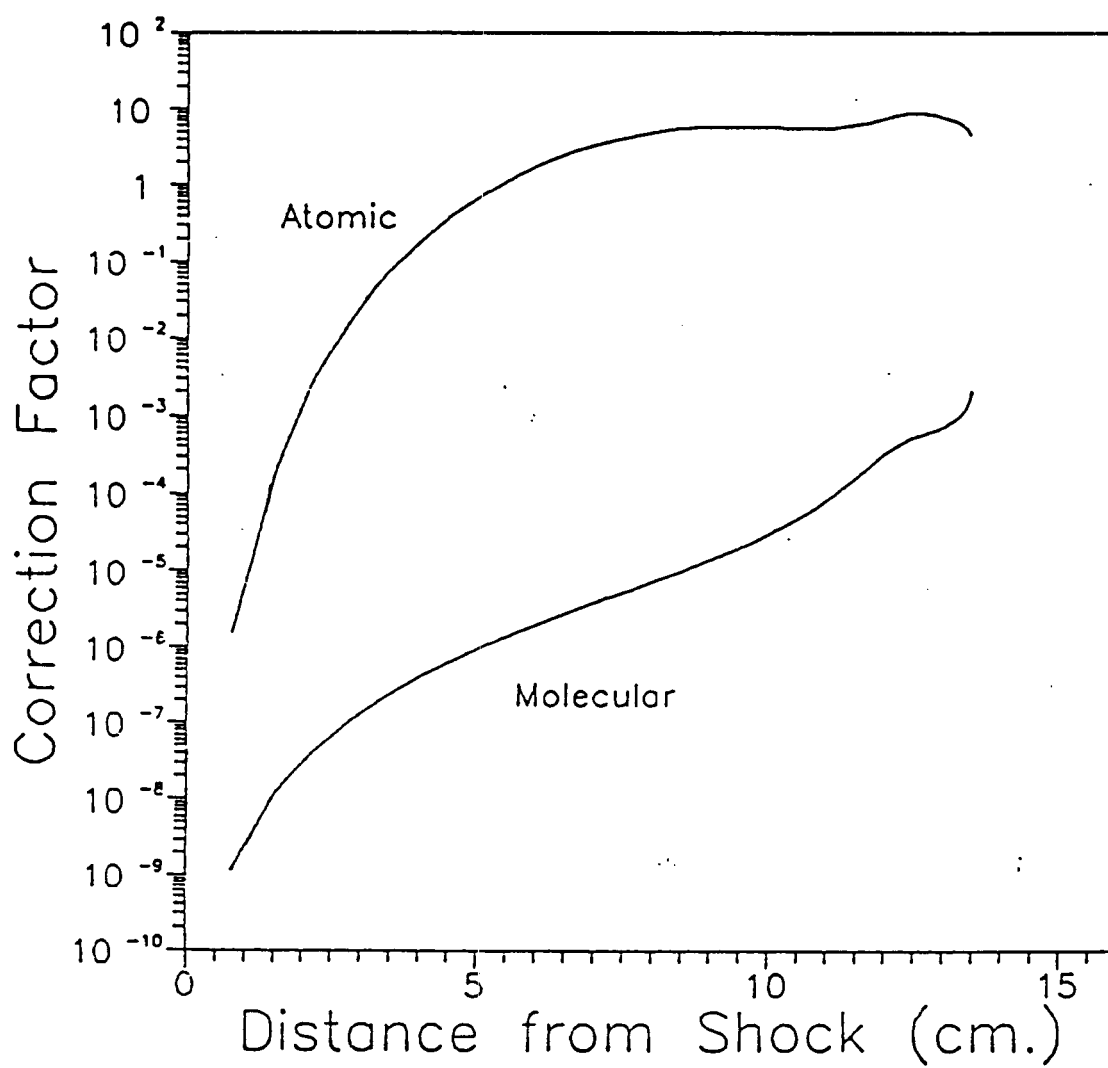


Figure 23. Nitrogen Correction Factors for Kang and Dunn Rate with CPL Coupling

model used in the modified Grose code.

Unfortunately, the Anderson model, while simple, predicts values of radiative flux that are not in agreement with the other step models. Apparently, two steps are insufficient to properly describe the radiative processes in the AFE flight regime, particularly when nonequilibrium effects are important.

The radiance model consistently predicts radiative heat transfers greater than those of the various step models. Since the density of the air at 80 km. is high enough that a large amount of self-absorption should take place, the radiance model, being a transparent gas model, is believed to also be inappropriate for the AFE flight regime.

If the integral in equation (66) for the radiative heat transfer to the wall is represented as the summation over the code points between the shock and the wall,

$$q_r = \sum \Delta q_r \quad (94)$$

it can be seen on Figure 24 that the high initial T_e predicted by the uncorrected CVDV model leads to the large predicted radiative heat transfer to the wall in the optically thin shock layer at 80 km. altitude. The large correction factors calculated for the Kang and Dunn rate with a corrected CVDV coupling model combined with the high peak T_e predicted behind the shock front, leads to a large amount of the wall radiative transfer originating from the gas close to the shock as seen in Figure 25. By contrast, Figure 26 shows that the wall radiative heat transfer predicted by the corrected CPL model using the Kang and Dunn rate primarily comes from the region close to the AFE body. Since the actual electron temperature is believed to

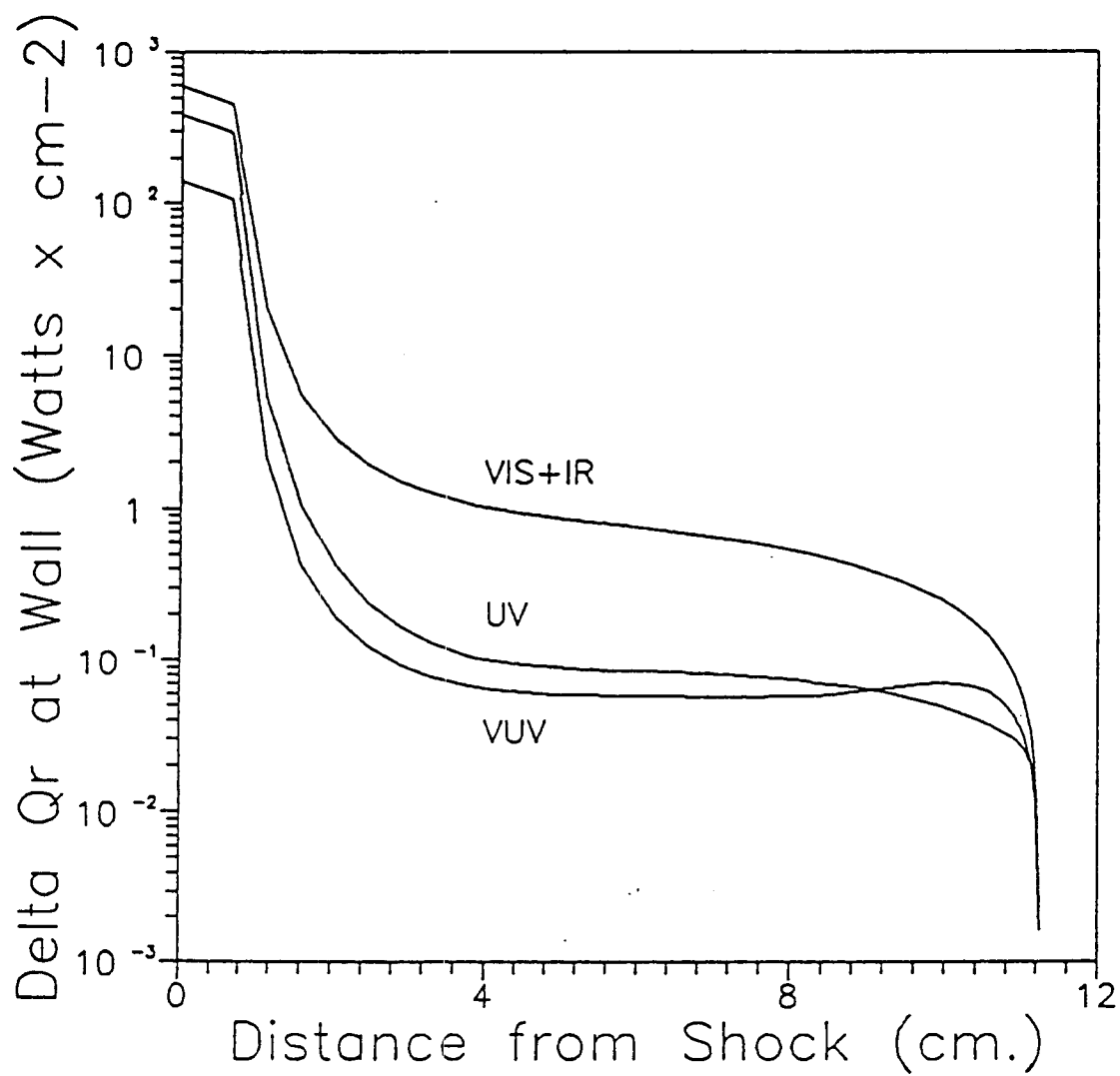


Figure 24. Incremental Radiative Heat Transfer to the Wall from Uncorrected CVDV Model Using Kang and Dunn Rate at $X = 9.0$ cm.

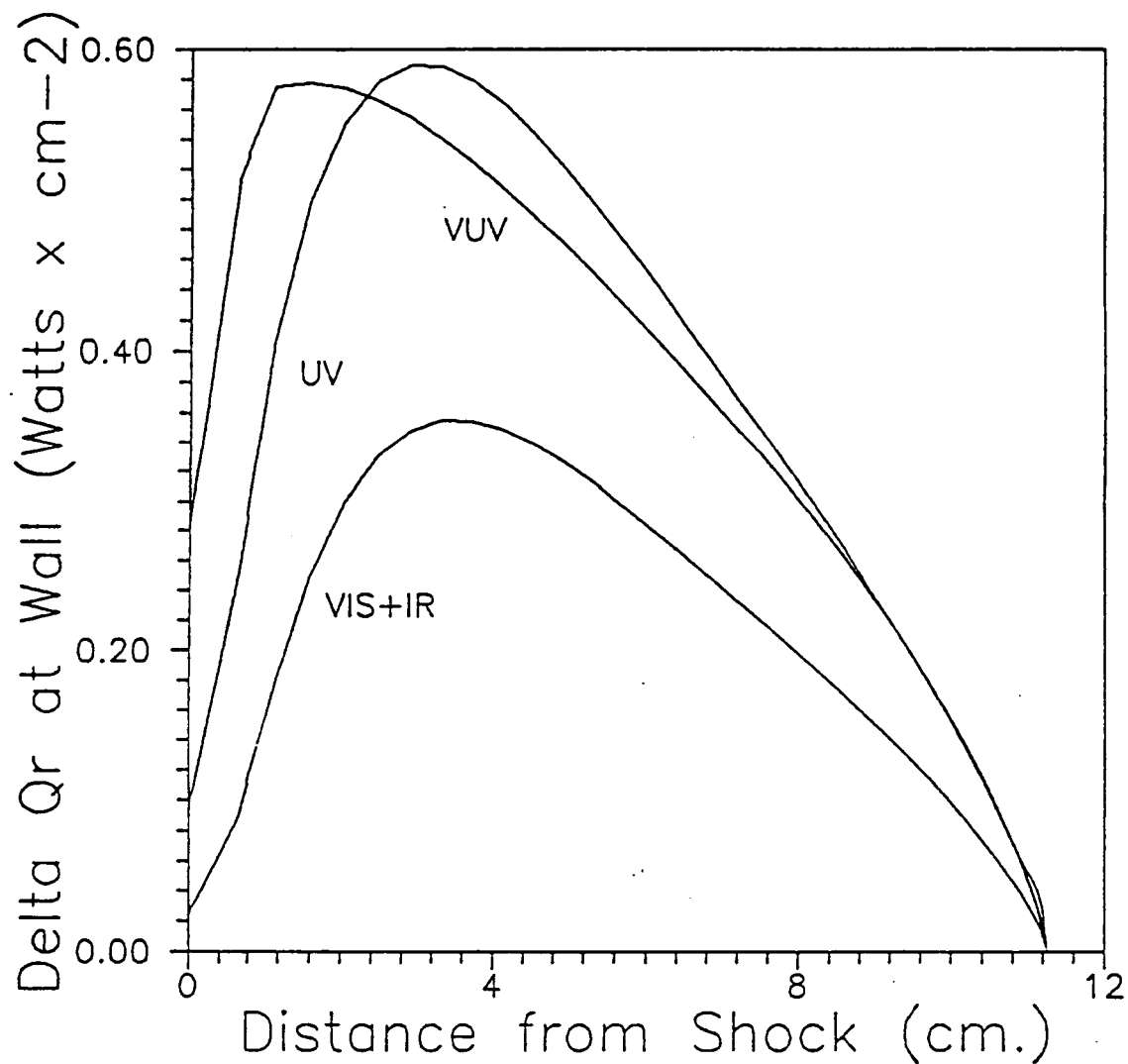


Figure 25. Incremental Radiative Heat Transfer to the Wall from Corrected CVDV Model Using Kang and Dunn Rate at $X = 9.0$ cm.

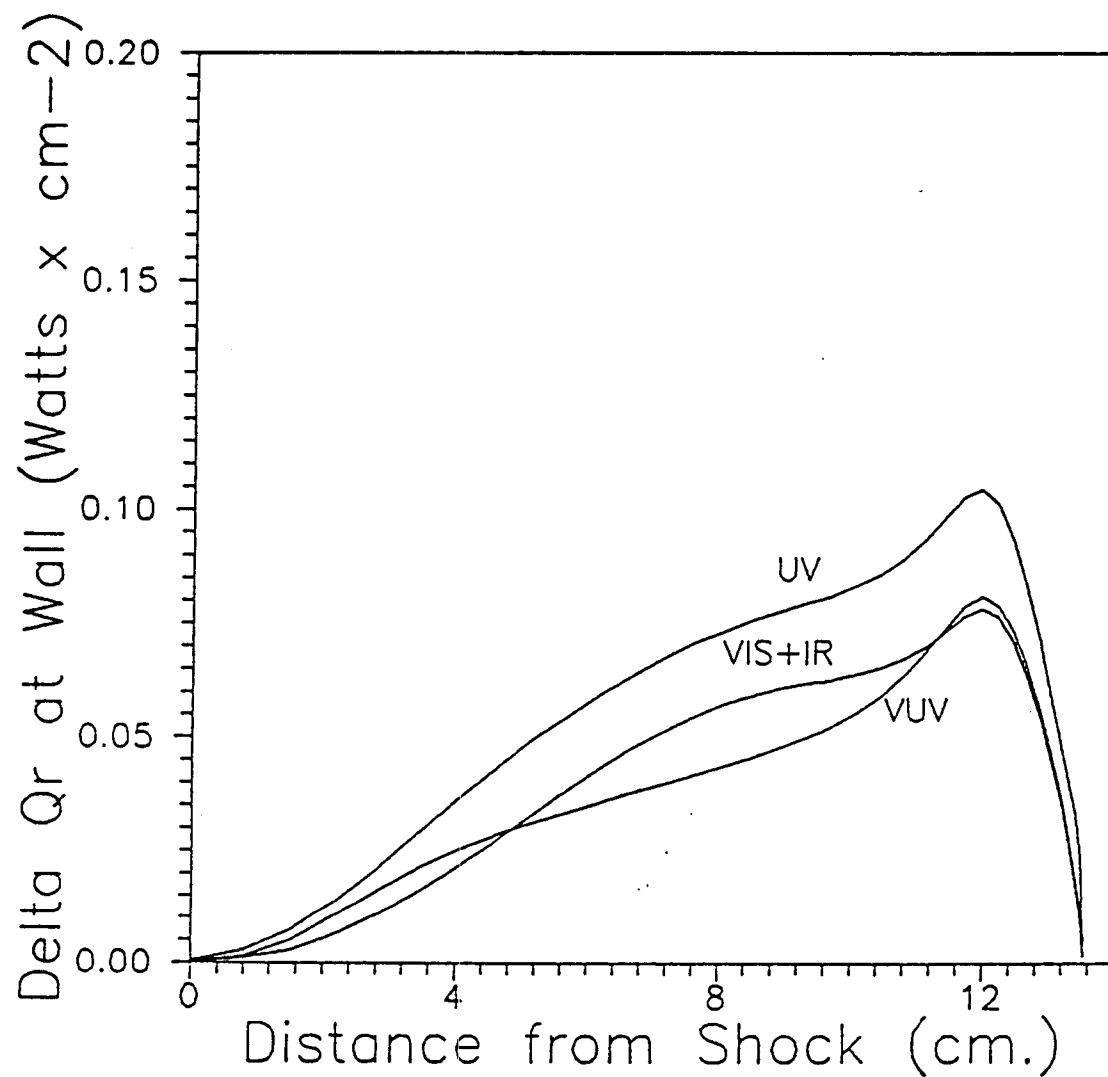


Figure 26. Incremental Radiative Heat Transfer to the Wall from Corrected CPL Model Using Kang and Dunn Rate at $X = 9.0$ cm.

have a lower initial increase and peak similar to that displayed by the CPL model, it is felt that the CPL model should provide a more accurate prediction of the radiative heat transfer than the other models.

The final problem then, is to choose between the Olstad and the KCN models for the more accurate radiative transfer model. While KCN and Olstad agree in the visible and infrared, they disagree in the vacuum ultraviolet, primarily due to choices of band widths. Since Olstad has band widths based upon the actual ionization edges of oxygen and nitrogen, it is probably the better of the two. In addition, the KCN model was originally derived for atomic nitrogen and correlated to air, and Table 7 shows that there is significant molecular radiation that is not taken into account by the KCN model. Further, Figures 20 - 23 indicate by the size of the molecular correction factors that there is a considerable LTNE effect present in the AFE flowfield. Consequently, the most accurate combination of the various radiative models, chemical rates, and vibration - dissociation coupling models is believed to be the Olstad model with molecular and atomic correction factors using the Kang and Dunn chemical reaction rates with CPL modelling. However, for a conservative estimate, it is suggested that the Kang and Dunn reaction rates, combined with the Olstad and CVDV models, would provide a reliable "worst case" estimate.

CONCLUSION

The shock jump chemistry models developed in this thesis yielded accurate results for the AFE flowfields investigated and increased the computational speed of the modified Grose code. However, the internal tolerances of the Grose code need further investigation in order to determine why several streamlines required more computational steps to obtain a solution with the oxygen equilibrium dissociation model than with frozen chemistry at the shock front. In addition, the nitrogen and oxygen equilibrium model needs to be investigated at higher freestream velocities than those associated with the AFE trajectory.

The assumption that $T_e = T_{vN_2}$ yielded the pattern expected for the electron temperature when used with the CPL vibration - dissociation coupling model. However, the assumption resulted in high electron temperatures near the shock front when used with the CVDV vibration - dissociation coupling model. The Carlson and Rieper correlation yielded reliable electron temperatures at low freestream velocities, but gave results that were too low at the AFE flight conditions. If the Carlson and Rieper correlation could be modified to account for the low number density of diatomic nitrogen in the AFE flowfield, it is believed that the correlation would give fast and accurate results for the AFE flight regime.

The radiative heat transfer was found to be highly dependent on the chemical reaction rates, vibration - dissociation coupling model, and the correction factors used. The best results were obtained with the Kang and Dunn electron impact ionization rates with CPL coupling and both

atomic and molecular correction factors included. The radiance model was found to yield radiative heat transfers that were too high since there was significant self-absorption in the AFE flowfield and the transparent gas assumption is inaccurate. The Anderson model was originally derived for electron temperatures that were well below those present throughout most of the AFE's flowfield and is believed to be inadequate for the present conditions of interest. The KCN model yielded results similar to the Olstad model; yet the KCN model was derived for atomic nitrogen, and it was felt that the Olstad model more accurately captured the effects of atomic and molecular band radiation. However, the radiative cross-sections for vacuum ultraviolet line regions in the Olstad model need to be refined for the higher altitudes of the AFE flight regime.

The LTNE correction factors in this thesis helped to adapt the radiative heat transfer models to the AFE flight regime, and the results yielded by the radiation models using the correction factors corresponded to those obtained by other methods^{3,18}. However, the assumption in the derivation of the molecular correction factors that the upper electronic levels of N_2 are populated in equilibrium with atomic nitrogen needs further validation.

In conclusion, various engineering models of several important aspects of the hypersonic flow about an AFE/AOTV type of vehicle have then been investigated. Based on these studies, it is believed that accurate flowfield and radiative heating results can be obtained by using together, the eight step Olstad absorption coefficient model, molecular and atomic correction factors, the Kang and Dunn electron impact nitrogen ionization rate expression, and the CPL vibration - dissociation coupling model. For a conservative estimate, the CPL model could be

replaced with the CVDV model to obtain reliable "worst case" estimates.

REFERENCES

1. Lee, J. H., "Basic Governing Equations for the Flight Regimes of Aeroassisted Orbital Transfer Vehicles," AIAA Paper 84-1729, June 1984.
2. Carlson, L. A. and Rieper, R. G., "Electronic Temperature and Relaxation Phenomena behind Shock Waves," Journal of Chemical Physics, Vol. 57, Number 2, July 1972, pp. 760-766.
3. Park, C., "Assessment of Two-Temperature Kinetic Model for Ionizing Air," AIAA Paper No. 87-1574, June 1987.
4. Curtiss, J. T. and Strom, C. R., "Computations of the Nonequilibrium Flow of a Viscous, Radiating Fluid about a Blunt Axisymmetric Body," AFFDL-TR-67-40, Vol. 1, Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio, June 1967.
5. Chapin, C. E., "Nonequilibrium Radiation and Ionization in Shock Waves," Ph.D. Dissertation, Purdue University, 1967.
6. Maslen, S. H., "Inviscid Hypersonic Flow Past Smooth Symmetric Bodies," AIAA Journal, Vol. 2, No. 6, June 1964, pp. 1055-1061.
7. Grose, W. L., "A Thin-Shock-Layer Solution for Nonequilibrium, Inviscid Hypersonic Flows in Earth, Martian, and Venusian Atmospheres," NASA TN D-6529, 1971.

8. Hammerling, P., Teare, J. D., and Kivel, B., "Theory of Radiation from Luminous Shock Waves in Nitrogen," Physics of Fluids, Vol. 2, No. 4, July-Aug. 1959, pp. 422-426.

9. Bobskill, G. J., "A Comparative Study of Reaction Rate, Species, and Vibration-Dissociation Coupling Models for an AOTV Flowfield," Master's Thesis, Texas A&M University, August 1988.

10. Vincenti, W. G. and Kruger, C. H., Introduction to Physical Gas Dynamics, Krieger Publishing Co., Malabar, Florida, 1965, pp. 152-156.

11. Lee, J. H., "Electron-Impact Vibrational Excitation Rates in the Flowfield of Aeroassisted Orbital Transfer Vehicles," Thermophysical Aspects of Re-Entry Flows: Progress in Astronautics and Aeronautics, Vol. 103, edited by J. D. Moss and C. D. Scott, American Institute of Aeronautics and Astronautics, New York, NY, 1986, pp. 197 - 224.

12. Nardone, M. C., Briene, R. G., Zeldin, S. S., and Riethoff, T. R., "Radiance of Species in High Temperature Air," General Electric Corp. Missile and Space Division Rept. No. R63SD3, 1963.

13. Allen, R. A., Rose, P. H., and Camm, J. C., "Non-Equilibrium and Equilibrium Radiation at Super-Satellite Re-Entry Velocities," IAS Paper No. 63-77, 1963.

14. Olstad, W. B., "Nongray Radiating Flow about Smooth Symmetric Bodies," AIAA Journal, Vol. 9, No. 1, January 1971, pp. 122 - 130.

15. Knott, P. R., Carlson, L. A., Nerem, R. M., "A Further Note on Shock-Tube Measurements of End-Wall Radiative Heat Transfer in Air," AIAA Journal, Vol. 7, No. 11, November 1969, pp. 2170 - 2172.
16. Anderson, J. D., "Heat Transfer from a Viscous Nongray Radiating Shock Layer," AIAA Journal, Vol. 6, No. 8, August 1968, pp. 1570 - 1573.
17. Fay, J. A., and Kemp, N. H., "Theory of Stagnation Point Heat Transfer in a Partially Ionized Diatomic Gas," AIAA Journal, Vol. 1, No. 12, December 1963, pp. 2741 - 2751.
18. Moss, J. N., Bird, G. A., and Dogra, V. K., "Nonequilibrium Thermal Radiation for an Aeroassist Flight Experiment Vehicle," AIAA Paper No. 88-0081, January 1988.
19. Kang, S. W., and Dunn, M. G., "Theoretical and Measured Electron-Density Distributions for the RAM Vehicle at High Altitudes," AIAA Paper No. 72-689, June 1972.
20. Wilson, J., "Ionization Rate of Air Behind High Speed Shock Waves," The Physics of Fluids, Vol. 9., No. 10, October 1966, pp. 1913-1921.
21. Carlson, L. A., "Radiative-Gasdynamic Coupling and Nonequilibrium Effects behind Reflected Shock Waves," AIAA Journal, Vol. 9, No. 5, May 1971, pp. 858-865.
22. Carnicom, M. L., "Reaction Rates for High Temperature Air," Sandia Corp. Report No. SC-R-66-885, May 1966.

VITA

Robert Brian Greendyke was born on [REDACTED] at [REDACTED]. The son of an Air Force officer, he travelled extensively throughout the United States and overseas. After graduating from Baylor University in Waco, Texas with a Bachelor's degree in economics and business administration, he briefly attended graduate school at the University of Washington in economics. Upon leaving graduate school, He worked as an Industrial Engineer for the Boeing Commercial Aircraft Company's 747 Division. Mr. Greendyke left the Boeing Co. to return to school in aerospace engineering at Texas A&M University. After obtaining his bachelor's in aerospace engineering in 1986, he attended graduate school at Texas A&M University, receiving his Master's in August, 1988. Mr. Greendyke is currently employed by Analytical Services and Materials, Inc. of Hampton, Virginia.

APPENDIX 5

Program RAPROX

An Approximate Solution for Hypervelocity, Nonequilibrium Radiating, Reacting,
and Conducting Stagnation Regions

Developed by

Leland A. Carlson

***** PROGRAM RAPROX *****

Program RAPROX is an easily used and rapid computer program designed to provide approximate stagnation point solutions for flowfield structure and estimates of radiative transfer for hypervelocity vehicles in the flight regime of 11-18 km/sec. It includes the effects of chemical and thermal nonequilibrium and nonequilibrium nongray radiative transfer. It is valid only when the flowfield is dominated by atomic and ionization processes and dissociation can be reasonably assumed to be "complete" in the shock front.

The program is based upon the theory presented in AIAA Paper No. 88-2672. However, some minor changes have been incorporated into the program since the publication of that paper. The radiation losses from the flowfield are still estimated using using specific bands of the 5-step model referenced. It is believed that this approach will give reasonable estimates of the losses. However, the radiative heating to the wall is now computed only with a modified version of the Olstad 8-step model. The original Olstad model has been simplified to only include nitrogen atom radiation. Also, the cross-sections for the 1130-1801 Angstrom wavelength region have been reduced by a factor of 2.5. This region was denoted as bands five and six in the Olstad model and was renumbered as bands six and seven in the AIAA paper. In addition, correlations for the adiabatic shock standoff distance and the chemical relaxation length have been built into the program as well as a routine to predict the theoretical equilibrium conditions. These changes simplify the usage of the code..

The program is written in Microsoft BASIC and is designed to operate on an AT compatible microcomputer under the Microsoft QuickBasic Compiler. (Note: The word "AT" may be a registered trademark of the IBM Corporation. The word "QuickBasic" is a registered trademark and a copyrighted computer software program from the Microsoft Corporation.) In its present form it cannot be operated under standard BASICA or GWBASIC since it contains many lines without statement numbers. However, the program could be easily modified for operation under these systems. The program also generates a graphical output and requires at least a CGA color graphics compatible system. While some of the output is in color, it could be viewed using a compatible monochrome system.

All output from the program is either to the screen or to files on the drive where the program is located. The files consist of a flowfield output file assigned a name selected by the user (for example SAMP), a summary and heat transfer data file denoted by *.HT (example SAMP.HT), two data files suitable for use with an external graphics program designated *.P1 and *.P2, and a plot image file *.BAS where * is an input name from the user (example SAMPP.BAS). The plot image file should be retrievable using the program PLOTPIC.BAS. The latter is listed in this appendix. Due to differences in systems, the external graphics packages for making plots from data has to be supplied by the user. Depending upon the packages used, the output data files *.P1 and *.P2 format(s) may have to be modified.

USER INSTRUCTIONS

In general RAPROX is a user oriented program and simple to use. The sample case contained in this appendix corresponds to the following input conditions: Freestream velocity 16 km/sec, Altitude of 75 km with a freestream pressure of 24.897 dynes/sq cm and a temperature of 200.15 K, Nose radius of 230 cm. In these instructions it will be assumed that the user has installed the program under QuickBasic and knows how to run the program under QuickBasic.

After giving the command to "Run" and "Start", the user will interactively be asked "U INFINITY IN KM/SEC IS ?". Enter 16 followed by return. Next "P INFINITY IN DYNES/SQ CM IS?" will appear. Enter 24.897 (Return). This will be followed by "T INFINITY IN DEG K IS ?" Input 200.15 (return). Subsequently the nose radius will be requested via "NOSE RADIUS IN CM IS ?". For the sample case enter 230.

At this point there will appear on the screen the iteration history of the routine which computes the ideal equilibrium conditions corresponding to this case. These conditions assume complete dissociation. On the screen will appear values for E2, the ideal equilibrium degree of ionization, and T2, the corresponding equilibrium temperature. For the sample cases the values are E2 = 0.627 and T2 = 13053.

Next the code will request values for IM and DTAU. IM is the maximum number of grid points to be used in both the thick and the thin regions. At present the code is dimensioned for maximum value of IM of 100. Typical practical values of IM are 50 and 100. For the sample case enter 100. DTAU is the increment to be used in the thick region. Appropriate values have to be selected fundamentally on experience. However, a little later the code will make an estimate of an appropriate DTAU and the user will have an opportunity to change the initially selected value. The "criterion" for an appropriate value is that in the final solution, the thick nonequilibrium region transitions "smoothly" into the thin equilibrium zone. If DTAU is selected too large, the thick zone may exceed the shock layer thickness or, more commonly, have a long tail of constant properties. If DTAU is too small, the thick zone behind the shock will be short and the transition to the equilibrium thin region will be marked by a sharp jump or discontinuity. For initial cases, the user will probably have to run a case a few times with various values of DTAU until he/she is "comfortable" physically with the results. For the sample case enter for DTAU a value of 0.075. Thus the IM and DTAU input should be 100,.075 followed by a return.

Next the user will be asked to supply a plot name. For the sample case use SAMPP. The additional P at the end of the name is suggested as an indicator that the file is a plot file. This input will be followed by a request for a name for an OUTPUT file. For the sample case use SAMP.

At this point the names and values for fifteen variables will appear on the screen, and the user will be asked whether or not these are acceptable. These variables are U1, the freestream velocity in cm/sec; P1, the freestream pressure; T2, the computed equilibrium temperature; E2, the computed equilibrium degree of ionization; S1,S2, B1, B2, computed radiative cross sections and source functions

used to compute cooling in the thick and thin zones; XR, the computed estimated chemical relaxation distance; L, the shock standoff distance corresponding to the fully dissociated nonreacting nonradiating case; N, an exponent used in the radiation term in the thin region; T1, the freestream temperature; TW, the wall temperature; PLOT TMAX, the maximum temperature ordinate for the screen plot when the computation is finished; and PLOT EPS MAX, the maximum ordinate for degree of ionization for the screen plot. In general, variables 5 - 11 would not be changed and are simply presented for information purposes. Variables 1, 2, and 12 might be changed if they were inputted incorrectly at the beginning. Variables 14 and 15 might be changed at the user's discretion to make the final screen plot more desirable. However, in general, the variable most likely to be changed is number 13, TW, the wall temperature; but it should be remembered that TW should be relatively high since the model assumes that the gas is composed only of atoms, ions, and electrons (no molecules). Also, if U1 is changed, remember that at this point it must be input in units of cm/sec, not Km/sec.

If the user desires at this point to change a variable, enter N followed by return. The user then will be asked "WHICH ONE SHOULD BE CHANGED?" Enter the number of the variable which is to be changed. Then, when prompted, enter the new value for the variable. The new values will then be displayed. Note that if freestream conditions are changed that the values for T2 and E2 are not changed. (Ideally they should be, but the code has not been modified to do this.) When the user is satisfied with the values, input Y (for yes) followed by return.

At this point a Reynolds number based upon conditions behind the shock and the reference length will appear followed by an estimate for DTAU and an inquiry concerning whether or not the user wishes to change DTAU. For the sample case the suggested value is 0.08 which is quite close to the inputted value of 0.075. Thus, N for no change.

At this point the actual flowfield solution will be computed and some numbers will appear and scroll on the screen. These numbers are associated with the iterative thin solution and simply indicate that the program is working. After the flowfield solution is obtained, it will be written to the appropriate output file, and a graphical plot of the temperature and degree of ionization will appear on the screen. At the top of the screen also will appear the estimated convective heating.

On the plot, the shock front is on the left while the wall is on the right. The red line is temperature and its scale, running from 0 to TMAX, is on the left. The green line is the degree of ionization and its scale, running from 0 to EPS MAX is on the right. The temperature in the thick chemical nonequilibrium zone is plotted as a solid line, while that for the thin equilibrium zone is plotted with dots for each point. However, sometimes the dots are close together and the line appears solid; but usually the location of the transition from the thick to the thin zone can be easily detected. If this transition is "smooth" and the "constant" zone at the end of the thick region is not too long, then DTAU has been selected appropriately. If these features are not present, then the case should be run again with a different DTAU value. It should also be noted that often for low TW values that the degree of ionization near the wall goes to zero. This trend would

be physically expected and doesn't seem to affect the quality of this approximate solution.

At this point the program proceeds to compute the radiative heating to the wall. This takes several seconds. When it is finished, there will appear at the top of the screen a summary of the radiative heating results, along with the shock standoff distance, XS, accounting for chemistry, radiative cooling, and heating to the wall. This output for the sample case is included in this appendix.

After storing the radiative heating results and the screen plot in the appropriate files, the program then asks if another calculation is desired. If yes enter Y and continue. If no, enter N and the program will return to QuickBasic or the system as appropriate.

To retrieve a plot, use the program PLOTPIC. In PLOTPIC, statement 40 may have to be modified to reflect on which drive the plot has been actually stored. To use it, unload any other program, load PLOTPIC, and then do a run and start from QuickBasic. When the prompt FILE NAME? appears enter the fundamental name of the file. For the sample case, enter SAMPP. The plot will appear along with a prompt "?". If you wish to retrieve another plot, enter Y. If not, enter N.

STATEMENT LISTING

```

1  REM ***** PROGRAM RAPROX *****
  REM
  REM   An approximate solution which includes the effects of chemical
  REM   and thermal nonequilibrium and nonequilibrium nongray radiative
  REM   transfer and which is suitable for rapidly obtaining approximate
  REM   stagnation point solutions for flowfield structure and estimates
  REM   of radiative heat transfer. Valid only when flowfield is dominated
  REM   by atomic and ionization processes and dissociation can be reasonably
  REM   assumed to be "complete" in the shock front. Thus, freestream
  REM   velocities should be in the 11 - 18 km/sec range.
  REM
  REM   Developed by Dr. Leland A. Carlson, Aerospace Engineering Dept.
  REM   Texas A&M University, College Station, TX 77843.
  REM   Program follows the theory presented in
  REM   AIAA Paper No. 88-2672.
  REM
9  DEF FNBB (VT) = (EXP(-VT) * (6 + 6 * VT + 3 * VT ^ 2 + VT ^ 3) + .5 * EXP(-2 * VT) * (.75 + 1
  .5 * VT + 1.5 * VT ^ 2 + VT ^ 3)) * .1539897#
10 DIM QR(8), TU(8), K(8), BO(8), KO(8), EI(8), EO(8), B(8)
11 DIM DA(20), NA$(20)
12 DIM H1(101), H2(101), BB(101), DD(101), CC(101), AA(101), X1(101), X2(101)
39 INPUT "U INFINITY IN KM/SEC IS "; U1
  U1 = U1 * 100000!
  INPUT "P INFINITY IN DYNES/SQ CM IS "; P1
  INPUT "T INFINITY IN DEG K IS "; T1
  INPUT "NOSE RADIUS IN CM IS "; RN
  HA = 3.36E+11; HI = 1.34E+12
  REM P1 = P1 / 760 * 1013000!
  HS = .5 * U1 * U1 - HA
  CP = 1.48E+07
  TS = HS / CP
  HA = HA / HS; HI = HI / HS
  R = 8.313E+07 / 28!
  R1 = P1 / (R * T1)
  PS = R1 * U1 * U1
  REM INPUT "E2"; E2
  E2 = .1
  FOR IT = 1 TO 50
    T2 = TS * (1 - E2 * (HI - HA)) / (1 + E2)
    D2 = -TS * (HI - HA) / (1 + E2) - TS / (1 + E2)
    FE = E2 * E2 / (1 - E2 ^ 2) - 1.498 * (T2 ^ 2.5) / PS * EXP(-169000 / T2)
    FP = 2 * E2 / (1 - E2 ^ 2) - E2 ^ 2 / ((1 - E2 ^ 2) ^ 2) - 1.498 * 2.5 / PS * (T2 ^ 1.5) * D2
    * EXP(-169000 / T2) - 1.498 * (T2 ^ .5) / PS * 169000 * D2 * EXP(-169000 / T2)
    EN = E2 - FE / FP
    IF EN < 0 THEN EN = E2 / 2
    ER = ABS(EN - E2)
    IF ER < .00001 GOTO 98
    E2 = EN
    PRINT "E2="; E2; " T2="; T2; " F="; FE; " FP="; FP
  NEXT IT
  PRINT "DIDN'T CONVERGE"
  GOTO 9999
98 PRINT "CONVERGED E2="; EN; " T2="; T2
  PRINT "IT="; IT
99 PRINT
21 GOSUB 2000
25 T3 = T2
26 GOSUB 3000
  UT = U1 / 100000!
  IF UT > 11 GOTO 27

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

XR = -.7782 * (UT - 10) + 5.5086
GOTO 40
27 IF UT > 12 GOTO 28
XR = -.5191 * (UT - 11) + 4.7305
GOTO 40
28 IF UT > 13 GOTO 29
XR = -.4366 * (UT - 12) + 4.2113
GOTO 40
29 IF UT > 14 GOTO 30
XR = -.3093 * (UT - 13) + 3.7747
GOTO 40
30 IF UT > 15 GOTO 31
XR = -.2603 * (UT - 14) + 3.4654
GOTO 40
31 IF UT > 16 GOTO 32
XR = -.1611 * (UT - 15) + 3.2051
GOTO 40
32 IF UT > 17 GOTO 33
XR = -.1105 * (UT - 16) + 3.044
GOTO 40
33 XR = -.0361 * (UT - 17) + 2.9335
40 XR = EXP(XR) / P1
N = 3
CP = 1.48E+07
MA = 2.32E-23
K = 1.38E-16
RR = PS * MA / K / TS / R1
L = .75 * RN / RR
TW = 1650
MT = 50000: EM = 1
HA = 3.36E+11
HI = 1.34E+12
R = (8.313E+07) / 28!
41 INPUT "IM AND DTAU"; IM, DT
42 INPUT "PLOT NAME"; PLS
43 INPUT "OUTPUT FILE NAME IS"; PLS
44 GOSUB 1000
50 PR = .67
55 HS = .5 * U1 * U1 - HA
60 TS = HS / CP
65 MU = (4! / 15!) * (MA / K) * 62.7 * (TS ^ .75)
70 R1 = P1 / (R * T1)
75 RE = R1 * U1 * L / MU
77 PRINT "REYNOLDS NO -"; RE
80 PS = R1 * U1 * U1
85 K1 = (PS / (K * TS)) * S1
90 A = K1 * L
95 K2 = S2 * (1 - E2) * PS / (K * (1 + E2) * T2)
100 G1 = 6.28 * B1 / (R1 * U1 * HS)
105 G2 = 12.57 * K2 * B2 * L / (R1 * U1 * HS)
106 E2 = (1 - T2 / TS) / (T2 / TS + (HI / HS - HA / HS))
110 H2 = (1 + E2) * T2 * CP / HS
D9 = (PS / (K * T2)) * S1 * XR / SQR(H2) / (IM - 1)
PRINT "ESTIMATED VALUE OF DTAU IS "; D9
PRINT "YOUR INPUT DTAU IS CURRENTLY "; DT
INPUT "DO YOU WISH TO CHANGE DTAU?"; AN$
IF AN$ = "N" GOTO 115
INPUT "NEW DTAU IS "; DT
115 TA = .5 * K1 * (1 + (1 / H2)) * XR
120 B = 2.31 / TA

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

125 HI = (1.34E+12) / HS
130 HA = (3.36E+11) / HS
135 A1 = (2.9E-12) * HS
136 E2 = (1 - T2 / TS) / (T2 / TS + (HI - HA))
139 BC = E2 - A1 * H2
140 SU = 0
141 FOR I = 2 TO IM
142 TI = (I - 1) * DT
143 SU = SU + .5 * DT * (EXP(-2 * TI) / (1 - TI / (K1 * L)) + EXP(-2 * (TI - DT)) / (1 - (TI - D
T) / (K1 * L)))
144 NEXT I
145 HB = (1 - G1 * SU - BC * (HI - HA)) / (1 + A1 * (HI - HA))
150 EQ = A1 * HB + BC
155 HE = HB / (1 + EQ)
160 C = (HI - HA) * EQ * B
165 D = HE * EQ * B * B
172 D1 = DT
175 FOR I = 1 TO IM
180 TI = (I - 1) * DT
185 BB(I) = A / (RE * PR * DT * DT) + (1 - TI / A) / (2 * DT)
190 DD(I) = -2 * A / (RE * PR * DT * DT)
195 AA(I) = A / (RE * PR * DT * DT) - (1 - TI / A) / (2 * DT)
200 CC(I) = G1 * EXP(-2 * TI) + ((1 - TI / A) * C - A * D / (RE * PR)) * EXP(-B * TI)
210 NEXT I
215 Z1 = -RE * PR * 2 * DT / A
220 Z2 = RE * PR * 2 * DT / A - 2 * DT * HE * EQ * B
225 DD(1) = DD(1) + BB(1) * Z1
230 AA(1) = AA(1) + BB(1)
235 CC(1) = CC(1) - BB(1) * Z2
240 BB(IM) = 0
241 AA(IM) = 0
242 DD(IM) = 1
243 CC(IM) = HB
245 IL = 1
250 IU = IM
255 GOSUB 9000
260 FOR I = 1 TO IM
265 H1(I) = CC(I)
270 NEXT I
275 SU = 0!
276 X1(1) = 0!
280 FOR I = 2 TO IM
285 SU = SU + .5 * (SQR(H1(I - 1)) + SQR(H1(I))) * DT / (K1 * L)
290 X1(I) = SU
295 NEXT I
300 H0 = (1 + EQ) * (H1(IM) - EQ * HE)
301 T3 = H0 / (1 + EQ) * TS
302 GOSUB 3000
303 K2 = S2 * (1 - EQ) * PS / (K * (1 + EQ) * T3)
304 G2 = 12.57 * K2 * B2 * L / (R1 * U1 * HS)
305 PRINT "HB="; HB; " H0="; H0
310 TM = K2 * L * SQR(H0)
312 NM = 1 - (IM - 1) * D1 / K1 / L
314 TB = TM * (1 - NM)
315 DT = (TM - TB) / (IM - 1)
325 TW = TW / TS
326 TI = (IM - 1) * D1
327 E = EQ * (1 - EXP(-B * TI))
328 BC = E - A1 * H0
330 HW = TW * (1 + BC) / (1 - A1 * TW)

```

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

```
335 C1 = 1 + A1 * (HI - HA)
340 FOR I = 1 TO IM
342 TI = (I - 1) * DT + TB
345 H2(I) = (HW - H0) / (TM - TB) * (TI - TB) + H0
350 NEXT I
352 IT = 0
354 IT = IT + 1
355 DD(1) = 1
360 AA(1) = 0
365 CC(1) = -(H2(1) - H0)
370 DD(IM) = 1
375 BB(IM) = 0
380 CC(IM) = HW - H2(IM)
385 FOR I = 2 TO IM - 1
387 TI = (I - 1) * DT + TB
390 BB(I) = -C1 * (1 - TI / TM) / (2 * DT) - TM / (RE * PR * DT * DT)
395 DD(I) = 2 * TM / (RE * PR * DT * DT) + G2 / (K2 * L) * N / H0 * ((H2(I) / H0) ^ (N - 1))
400 AA(I) = C1 * (1 - TI / TM) / (2 * DT) - TM / (RE * PR * DT * DT)
405 CC(I) = BB(I) * H2(I - 1) + 2 * TM / (RE * PR * DT * DT) * H2(I) + G2 / (K2 * L) * ((H2(I) /
H0) ^ N) + AA(I) * H2(I + 1)
410 CC(I) = -CC(I)
415 NEXT I
420 IL = 1
425 IU = IM
430 GOSUB 9000
435 FOR I = 1 TO IM
440 H2(I) = H2(I) + CC(I)
445 NEXT I
450 DM = 0
455 FOR I = 1 TO IM
460 T5 = ABS(CC(I))
465 IF T5 < DM GOTO 475
470 DM = T5
475 NEXT I
480 IF DM < .001 GOTO 520
485 FOR I = 1 TO IM
490 PRINT CC(I); " ";
495 NEXT I
500 PRINT
505 IF IT > 5 GOTO 9500
510 GOTO 354
520 PRINT "THIN CONVERGED IN "; IT
525 SU = 0
530 X2(1) = X1(IM)
535 FOR I = 2 TO IM
540 SU = SU + .5 * (SQR(H2(I) / H0) + SQR(H2(I - 1) / H0)) * DT / (K2 * L)
545 X2(I) = SU + X2(1)
550 NEXT I
560 XS = X2(IM)
562 OPEN PL$ FOR OUTPUT AS #1
565 REM PRINT D$; "WRITE"; PL$
571 RR = PS * MA / K / TS / R1
572 PRINT #1, "THICK SOLUTION"
573 PRINT #1, "RE-"; RE; " G1-"; G1; " G2-"; G2; " U-"; U1; " P1-"; P1; " RHOS/RINF-"; RR
575 FOR I = 1 TO IM
580 TI = (I - 1) * DT
585 E = EQ * (1 - EXP(-B * TI))
590 T = H1(I) - E * HE
595 TD = T * HS / CP
596 RB = 1 / H1(I)
```



```

597 UB = -(1 - TI / A) * H1(I)
598 XL = (XS - X1(I)) * L
600 PRINT #1, "X-"; X1(I); " H-"; H1(I); " T-"; TD; " E-"; E; " R-"; RB; " U-"; UB; " X-"; XL
605 NEXT I
607 PRINT #1, "THIN SOLUTION"
610 FOR I = 1 TO IM
615 TI = (I - 1) * DT + TB
620 E = A1 * H2(I) + BC
625 T = H2(I) / (1 + E)
630 TD = T * HS / CP
631 RB = 1 / H2(I)
632 UB = -(1 - TI / TM) * H2(I)
633 XL = (XS - X2(I)) * L
635 PRINT #1, "X-"; X2(I); " H-"; H2(I); " T-"; TD; " E-"; E; " R-"; RB; " U-"; UB; " X-"; XL
640 NEXT I
645 CLOSE #1
646 GOSUB 1500
647 PL$ = PL$ + ".HT"
650 CLS
    SCREEN 1
651 COLOR 0, 0
652 LINE (0, 0)-(279, 0), 1
    LINE -(279, 159), 1
    LINE -(0, 159), 1
    LINE -(0, 0), 1
653 T = H1(1)
    MT = MT / TS
654 IF T > MT THEN MT = T
655 M1 = MT * TS
657 TP = -T / MT * 159 + 159: XP = 0
658 LINE (XP, TP)-(XP, TP), 2
660 FOR I = 1 TO IM
665 TI = (I - 1) * D1
670 E = EQ * (1 - EXP(-B * TI))
675 T = H1(I) - E * HE
690 TP = -T / MT * 159 + 159
695 XP = X1(I) / X2(IM) * 279
700 LINE -(XP, TP), 2
710 NEXT I
720 E = A1 * H2(1) + BC
725 T = H2(1) / (1 + E)
730 TP = -T / MT * 159 + 159
732 XP = X2(1) / X2(IM) * 279
735 LINE (XP, TP)-(XP, TP), 2
740 FOR I = 2 TO IM
745 E = A1 * H2(I) + BC
746 IF E < 0 THEN E = 0
750 T = H2(I) / (1 + E)
755 TP = -T / MT * 159 + 159
760 XP = X2(I) / X2(IM) * 279
765 LINE (XP, TP)-(XP, TP), 2
770 NEXT I
771 E5 = EM
772 IF 1.1 * E2 > EM THEN EM = 1.1 * E2
775 FOR I = 1 TO IM
780 TI = (I - 1) * D1
785 E = EQ * (1 - EXP(-B * TI))
790 EP = E / (EM) * (-159) + 159
795 XP = X1(I) / X2(IM) * 279
800 LINE (XP, EP)-(XP, EP), 1

```

ORIGINAL PAGE IS
OF POOR QUALITY

C-5

ORIGINAL PAGE IS
OF POOR QUALITY

```
810 E = A1 * H2(I) + BC
815 IF E < 0 THEN E = 0
820 EP = E / (EM) * (-159) + 159
825 XP = X2(I) / X2(IM) * 279
830 LINE (XP, EP)-(XP, EP), 1
835 NEXT I
840 T5 = 10000
845 TP = -T5 / TS / MT * 159 + 159
850 IF TP < 0 GOTO 870
855 LINE (0, TP)-(5, TP), 3
860 T5 = T5 + 10000
865 GOTO 845
870 E3 = .1
875 EP = E3 / (EM) * (-159) + 159
880 IF EP < 0 GOTO 900
885 LINE (275, EP)-(279, EP), 3
890 E3 = E3 + .1
895 GOTO 875
900 MT = M1
902 EM = E5
DEF SEG = &HB800
905 BSAVE PN$, 0, &H4000
907 OPEN PL$ FOR OUTPUT AS #1
908 REM PRINT D$; "WRITE"; PL$
912 FOR I = 1 TO 15 STEP 2
915 PRINT #1, NA$(I); "-"; DA(I); " "; NA$(I + 1); "-"; DA(I + 1)
920 NEXT I
921 PRINT #1, "E2="; E2; " EQ="; EQ
925 PRINT #1, "IM="; IM; " DTAU="; D1; " XS="; XS * L
PRINT #1, "RB="; RN
935 QR = 0
940 FOR I = 2 TO IM
945 QR = QR + .5 * ((H2(I) / H0) ^ N + (H2(I - 1) / H0) ^ N) * DT
950 NEXT I
955 QR = QR * 12.57 * B2 / 2 / 1E+07
960 REM PRINT #1, "QR="; QR; " WATTS/SQ CM"
961 QT = 12.57 * K2 * B2 * SQR(H0) / K1 * ((HE / H0) ^ N) * (IM - 1) * D1 / 2E+07
962 REM PRINT #1, "QRT="; QT; " WATTS/SQ CM"
963 QT = QT + QR
964 REM PRINT #1, "QR(TOTAL VIS)="; QT
970 GOSUB 7000
977 PRINT #1, "QC="; QC; " WATTS/SQ CM"
978 GOSUB 8300
979 CLOSE #1
981 PRINT "VUV = "; QR(1) + QR(2) + QR(3) + QR(4); " XS="; XS * L
982 PRINT "LINE REGION = "; QR(6) + QR(7); "RB="; RN
983 PRINT "VIS AND IR = "; QR(5) + QR(8); " QRTOT="; QR(1) + QR(2) + QR(3) + QR(4) + QR(5) + QR
(6) + QR(7) + QR(8)
985 GOTO 9990
1000 DA(1) = U1
1002 DA(2) = P1
1003 DA(3) = T2
1004 DA(4) = E2
1005 DA(5) = S1
1006 DA(6) = S2
1007 DA(7) = B1
1008 DA(8) = B2
1009 DA(9) = XR
1010 DA(10) = L
1011 DA(11) = N
```

```

1012 DA(12) - T1
1013 DA(13) - TV
1014 DA(14) - MT
1015 DA(15) - EM
1021 NA$(1) - "U1"
1022 NA$(2) - "P1"
1023 NA$(3) - "T2"
1024 NA$(4) - "E2"
1025 NA$(5) - "S1"
1026 NA$(6) - "S2"
1027 NA$(7) - "B1"
1028 NA$(8) - "B2"
1029 NA$(9) - "XR"
1030 NA$(10) - "L"
1031 NA$(11) - "N"
1032 NA$(12) - "T1"
1033 NA$(13) - "TW"
1034 NA$(14) - "PLOT TMAX"
1035 NA$(15) - "PLOT EPS MAX"
1060 CLS
1065 FOR I = 1 TO 15
1067 PRINT "VAR "; I; " "; NA$(I); "-"; DA(I)
1069 NEXT I
1075 INPUT "ARE THESE OKAY? Y OR N"; AN$
1080 IF AN$ = "Y" GOTO 1125
1085 IF AN$ = "N" GOTO 1095
1090 GOTO 1075
1095 INPUT "WHICH ONE SHOULD BE CHANGED? "; I
1100 PRINT "INPUT NEW VALUE FOR "; NA$(I)
1115 INPUT DA(I)
1116 T2 = DA(3)
1117 T3 = T2
1118 GOSUB 2000
1119 GOSUB 3000
1120 GOTO 1060
1125 U1 = DA(1)
1132 P1 = DA(2)
1133 T2 = DA(3)
1134 E2 = DA(4)
1135 S1 = DA(5)
1136 S2 = DA(6)
1137 B1 = DA(7)
1138 B2 = DA(8)
1139 XR = DA(9)
1140 L = DA(10)
1141 N = DA(11)
1142 T1 = DA(12)
1143 TW = DA(13)
1144 MT = DA(14)
1145 EM = DA(15)
1146 RETURN
1500 P1$ = P1$ + ".P1"
1505 OPEN P1$ FOR OUTPUT AS #1
1510 REM PRINT D$; "WRITE"; P1$
1515 PRINT #1, IM
1520 PRINT #1, IM
1522 FOR J = 1 TO 2
1525 FOR I = 1 TO IM
1530 TI = (I - 1) * D1
1535 E = EQ * (1 - EXP(-B * TI))

```

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

```
1540 T = H1(I) - E * HE
1545 TD = T * HS / CP / 10000
1550 XL = (XS - X1(I)) * L
1555 PRINT #1, XL
1560 IF J = 1 THEN PRINT #1, TD
1562 IF E < 0 THEN E = 0
1565 IF J = 2 THEN PRINT #1, E * 5
1570 NEXT I
1575 NEXT J
1580 CLOSE #1
1600 P2$ = PL$ + ".P2"
1605 OPEN P2$ FOR OUTPUT AS #1
1610 REM PRINT D$: "WRITE"; P2$
1615 PRINT #1, IM: PRINT #1, IM
1625 FOR J = 1 TO 2
1630 FOR I = 1 TO IM
1635 TI = (I - 1) * DT + TB
1640 E = A1 * H2(I) + BC
1645 T = H2(I) / (1 + E)
1650 TD = T * HS / CP / 10000
1655 XL = (XS - X2(I)) * L
1660 PRINT #1, XL
1665 IF J = 1 THEN PRINT #1, TD
1667 IF E < 0 THEN E = 0
1670 IF J = 2 THEN PRINT #1, E * 5
1675 NEXT I
1680 NEXT J
1685 CLOSE #1
1690 RETURN
2000 CF = 2.302585
2003 Y = T2 / 10000!
2005 IF T2 > 8000 GOTO 2025
2010 REM S1=8.17E-16
2011 S1 = 0
2015 B1 = 0
2020 GOTO 2170
2025 IF T2 > 10000 GOTO 2055
2035 S1 = (1.0061E-13) * (10! ^ (-2.6064 * Y))
2040 B1 = 10! ^ (-65.26966 + 15.87146 * LOG(T2) / CF)
2050 GOTO 2170
2055 IF T2 > 12000 GOTO 2085
2065 S1 = (9.6882E-15) * (10! ^ (-1.5953 * Y))
2070 B1 = 10! ^ (-54.67501 + 13.22279 * LOG(T2) / CF)
2080 GOTO 2170
2085 IF T2 > 14000 GOTO 2110
2090 S1 = (2.034E-15) * (10! ^ (-1.0304 * Y))
2095 B1 = 10! ^ (-47.32488 + 11.42093 * LOG(T2) / CF)
2105 GOTO 2170
2110 IF T2 > 16000 GOTO 2135
2115 S1 = (6.9685E-16) * (10! ^ (-.6981 * Y))
2120 B1 = 10! ^ (-41.90448 + 10.11359 * LOG(T2) / CF)
2130 GOTO 2170
2135 IF T2 > 18000 GOTO 2160
2140 S1 = (3.3188E-16) * (10! ^ (-.49675 * Y))
2145 B1 = 10! ^ (-37.77432 + 9.131181 * LOG(T2) / CF)
2155 GOTO 2170
2160 S1 = (2.0976E-16) * (10! ^ (-.38605 * Y))
2165 B1 = 10! ^ (-34.49941 + 8.36166 * LOG(T2) / CF)
2170 B1 = B1 * 1E+09
2172 DA(5) = S1
```

```

2175 DA(7) = B1
2180 RETURN
3000 CF = 2.302585
3005 Y = T3 / 10000!
3010 IF T3 > 8000 GOTO 3030
3015 S2 = 2.26E-24
3020 B2 = 0
3025 GOTO 3140
3030 IF T3 > 10000 GOTO 3050
3035 S2 = (3.4065E-30) * (10! ^ (7.2773 * Y))
3040 B2 = 10! ^ (-16.52511 + 4.425444 * LOG(T3) / CF)
3045 GOTO 3140
3050 IF T3 > 12000 GOTO 3070
3055 S2 = (8.3188E-28) * (10! ^ (4.8895 * Y))
3060 B2 = 10! ^ (-15.30031 + 4.119244 * LOG(T3) / CF)
3065 GOTO 3140
3070 IF T3 > 14000 GOTO 3090
3075 S2 = (2.8643E-26) * (10! ^ (3.6087 * Y))
3080 B2 = 10! ^ (-14.17392 + 3.843113 * LOG(T3) / CF)
3085 GOTO 3140
3090 IF T3 > 16000 GOTO 3110
3095 S2 = (1.9928E-25) * (10! ^ (3.00695 * Y))
3100 B2 = 10! ^ (-13.12 + 3.588921 * LOG(T3) / CF)
3105 GOTO 3140
3110 IF T3 > 18000 GOTO 3130
3115 S2 = (3.9376E-24) * (10! ^ (2.1971 * Y))
3120 B2 = 10! ^ (-12.13764 + 3.355253 * LOG(T3) / CF)
3125 GOTO 3140
3130 S2 = (4.2515E-24) * (10! ^ (2.1786 * Y))
3135 B2 = 10! ^ (-11.2815 + 3.154 * LOG(T3) / CF)
3140 B2 = B2 * 1E+10
3145 DA(6) = S2
3150 DA(8) = B2
3155 RETURN
7000 T4 = TW * TS
7005 VI = (4! / 15!) * (MA / K) * 62.7 * (T4 ^ .75)
7010 T3 = H0 / (1 + EQ) * TS
7015 GOSUB 3000
7020 K2 = S2 * (1 - EQ) * PS / (K * (1 + EQ) * T3)
7025 QC = VI * HS / PR * K2 / SQR(H2(IM) / H0) * (-2 * H2(IM - 3) + 9 * H2(IM - 2) - 18 * H2(IM - 1) + 11 * H2(IM)) / (6 * DT)
7030 QC = -QC / 1E+07
7035 PRINT "QC="; QC; " WATTS/SQ CM"
7040 RETURN
8300 PI = 3.14159; SB = 5.6697E-05
8305 FOR J = 1 TO 8: QR(J) = 0: EO(J) = 0: TU(J) = 0: NEXT J
8310 T = TW * TS: E = 0: NA = PS / (K * T)
8315 GOSUB 8800
8320 FOR J = 1 TO 8: KO(J) = K(J): BO(J) = B(J): NEXT J
8325 FOR I = IM - 1 TO 1 STEP -1
8330 E = A1 * H2(I) + BC: T = H2(I) / (1 + E) * HS / CP
8335 IF E < 0 THEN E = 0
8340 NA = (1 - E) * PS / ((1 + E) * K * T)
8345 GOSUB 8800
8350 FOR J = 1 TO 8
8355 TU(J) = TU(J) + .5 * (K(J) + KO(J)) * (X2(I + 1) - X2(I)) * L
8360 EI(J) = EXP(-2 * TU(J))
8365 QR(J) = QR(J) + .5 * (BO(J) * KO(J) * EO(J) + EI(J) * K(J) * B(J)) * (X2(I + 1) - X2(I)) * L
8370 BO(J) = B(J): KO(J) = K(J): EO(J) = EI(J)

```

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

```
8372 NEXT J
8375 NEXT I
8380 FOR I = 1M - 1 TO 1 STEP -1
8385 TI = (I - 1) * D1
8390 E = EQ * (1 - EXP(-B * TI))
8395 T = HE * HS / CP: TH = (H1(I) - E * HE) * HS / CP
8400 NA = (1 - E) * PS / (K * (TH + E * T))
8405 CR = E * E / (1 - E) * (1 - EQ) / (EQ ^ 2)
8410 GOSUB 8800
8415 FOR J = 1 TO 4: B(J) = CR * B(J): NEXT J
8420 FOR J = 5 TO 8: K(J) = CR * K(J): NEXT J
8425 FOR J = 1 TO 8
8430 TU(J) = TU(J) + .5 * (K(J) + KO(J)) * (X1(I + 1) - X1(I)) * L
8435 EI(J) = EXP(-2 * TU(J))
8440 QR(J) = QR(J) + .5 * (BO(J) * KO(J) * EO(J) + EI(J) * K(J) * B(J)) * (X1(I + 1) - X1(I)) *
L
8445 BO(J) = B(J): KO(J) = K(J): EO(J) = EI(J)
8447 NEXT J
8450 NEXT I
8455 FOR J = 1 TO 8: QR(J) = QR(J) * 6.2832 / 1E+07: PRINT #1, QR(J); " "; : NEXT J
8457 PRINT #1,
8458 PRINT #1, "VUV "; (QR(1) + QR(2) + QR(3) + QR(4))
8459 PRINT #1, "VUV LINES "; (QR(6) + QR(7))
8460 PRINT #1, "VIS+IR "; (QR(5) + QR(8)); " QRTOT="; QR(1) + QR(2) + QR(3) + QR(4) + QR(5) + Q
R(6) + QR(7) + QR(8)
8461 RETURN
8800 BT = SB * (T ^ 4) / PI
8805 TT = T / 168800: VT = 1 / TT
8810 B(1) = BT * (FNBB(VT))
8815 VT = .935 / TT: B(2) = BT * (FNBB(VT)) - B(1)
8820 VT = .835 / TT
8825 B(3) = BT * (FNBB(VT)) - B(2) - B(1)
8830 VT = .754 / TT
8840 B(4) = BT * (FNBB(VT)) - B(3) - B(2) - B(1)
8845 VT = .473 / TT
8850 B(6) = BT * (FNBB(VT)) - B(4) - B(3) - B(2) - B(1)
8855 VT = .213 / TT: B(8) = BT * (FNBB(VT))
8860 B(5) = BT - B(8) - B(6) - B(4) - B(3) - B(2) - B(1)
8865 NE = E * NA / (1 - E)
8870 B(7) = 2.4E-21 * NE * B(6) * EXP(.162 / TT)
8875 K(4) = 1.7E-17 * NA * EXP(-.246 / TT)
8880 K(3) = 2.1E-17 * NA * EXP(-.165 / TT) + K(4)
8885 K(2) = K(3)
8890 K(1) = 1.1E-17 * NA + K(2)
8895 K(5) = 2.6E-17 * NA * EXP(-.723 / TT)
8900 K(6) = 6E-18 * NA * EXP(-.379 / TT) + K(5)
8905 K(8) = 3.2E-17 * NA * EXP(-.631 / TT) + K(5)
8906 IF NE = 0 THEN K(7) = 0
8908 IF NE = 0 GOTO 8915
8910 K(7) = 1200 * NA / NE * EXP(-.489 / TT) + K(6)
8915 RETURN
8999 REM TRI-DIAGONAL SOLVER
9000 LP = IL + 1
9010 FOR I = LP TO IU
9020 RG = BB(I) / DD(I - 1)
9030 DD(I) = DD(I) - RG * AA(I - 1)
9040 CC(I) = CC(I) - RG * CC(I - 1)
9050 NEXT I
9060 CC(IU) = CC(IU) / DD(IU)
9070 FOR I = LP TO IU
```

9080 J = IU - I + IL
9090 CC(J) = (CC(J) - AA(J) * CC(J + 1)) / DD(J)
9100 NEXT I
9110 RETURN
9500 PRINT "ABNORMAL STOP"
9505 GOTO 9999
9990 INPUT "ANOTHER CALCULATION? Y OR N"; AN\$
9991 IF AN\$ = "N" GOTO 9998
9992 IF AN\$ = "Y" GOTO 9994
9993 GOTO 9990
9994 SCREEN 0
9995 GOSUB 1125
9996 GOTO 39
9998 BSAVE PN\$, 0, &H4000
9999 END

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

REM ***** PROGRAM PLOTPIC.BAS *****

```
5 DEF SEG = &HB800
10 SCREEN 0
24 CLS
25 INPUT "FILE NAME"; EXT$
30 SCREEN 1
36 COLOR 0, 0
40 A$ = "C:\QBASIC4\" + EXT$
41 PRINT A$
50 BLOAD A$, 0
51 FOR I = 1 TO 20
52 PRINT
53 NEXT
55 INPUT AN$
60 IF AN$ = "Y" GOTO 24
```

FLOWFIELD SOLUTION
Output from file SAMP

RE = Reynolds No., G1 and G2 are radiation cooling values. The X on the left hand side is nondimensional coordinate measured from the shock. The X on the right hand side is dimensional value in cm from the wall. T is translational temperature in deg K., E is degree of ionization, R is the density ratio ρ/ρ_0 , and U is nondimensional velocity u/U_0 . Note that the T immediately behind the shock assumes that dissociation is complete in the shock front and that immediately behind the shock the gas is composed entirely of atoms only.

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

THICK SOLUTION

RE- 151.9093 G1- 4.757869E-02 G2- 4.230994E-02 U- 1600000 P1- 24.897 RHOS/RINF- 6.747433
X- 0 H- .7940006 T- 50644.36 E- 0 R- 1.259445 U- .7940006 X- 14.47891
X- 2.303675E-03 H- .752504 T- 47305.43 E- 5.346203E-02 R- 1.328897 U- .7505324 X- 14.42002
X- 4.547589E-03 H- .7147815 T- 44267.88 E- .1022396 R- 1.399029 U- .711036 X- 14.36265
X- 6.735756E-03 H- .680485 T- 41504.18 E- .1467433 R- 1.46954 U- .6751364 X- 14.30671
X- 8.871973E-03 H- .6492992 T- 38989.37 E- .1873475 R- 1.540122 U- .6424946 X- 14.2521
X- 1.095983E-02 H- .6209384 T- 36700.82 E- .2243939 R- 1.610466 U- .6128042 X- 14.19872
X- .0130027 H- .5951437 T- 34617.95 E- .2581943 R- 1.680266 U- .5857881 X- 14.14649
X- 1.500379E-02 H- .5716802 T- 32722.13 E- .289033 R- 1.74923 U- .5611956 X- 14.09533
X- 1.696608E-02 H- .5503349 T- 30996.39 E- .3171696 R- 1.817075 U- .5387999 X- 14.04517
X- 1.889241E-02 H- .5309148 T- 29425.36 E- .3428409 R- 1.883541 U- .5183958 X- 13.99592
X- 2.078542E-02 H- .5132444 T- 27995.06 E- .3662628 R- 1.94839 U- .4997975 X- 13.94753
X- 2.264759E-02 H- .4971647 T- 26692.78 E- .3876325 R- 2.011406 U- .4828364 X- 13.89992
X- 2.448124E-02 H- .4825311 T- 25506.99 E- .4071297 R- 2.072405 U- .4673604 X- 13.85304
X- 2.628856E-02 H- .4692126 T- 24427.19 E- .4249186 R- 2.13123 U- .4532313 X- 13.80684
X- 2.807156E-02 H- .45709 T- 23443.85 E- .4411488 R- 2.187753 U- .440324 X- 13.76125
X- 2.983214E-02 H- .4460552 T- 22548.31 E- .4559568 R- 2.241875 U- .4285253 X- 13.71624
X- 3.157205E-02 H- .4360099 T- 21732.67 E- .4694674 R- 2.293526 U- .4177324 X- 13.67176
X- 3.329294E-02 H- .4268647 T- 20989.77 E- .4817941 R- 2.342663 U- .4078522 X- 13.62777
X- 3.499632E-02 H- .4185385 T- 20313.1 E- .4930408 R- 2.389266 U- .3988003 X- 13.58422
X- .0366836 H- .4109575 T- 19696.71 E- .503302 R- 2.433342 U- .3905001 X- 13.54108
X- 3.835608E-02 H- .4040546 T- 19135.22 E- .5126642 R- 2.474913 U- .3828822 X- 13.49833
X- 4.001499E-02 H- .3977689 T- 18623.71 E- .521206 R- 2.514023 U- .3758837 X- 13.45592
X- 4.166142E-02 H- .3920448 T- 18157.72 E- .5289993 R- 2.550729 U- .3694475 X- 13.41382
X- 4.329642E-02 H- .3868321 T- 17733.18 E- .5361098 R- 2.585101 U- .3635217 X- 13.37203
X- 4.492093E-02 H- .3820848 T- 17346.39 E- .5425973 R- 2.61722 U- .3580593 X- 13.33049
X- 4.653582E-02 H- .377761 T- 16993.98 E- .5485163 R- 2.647176 U- .3530178 X- 13.28921
X- 4.814192E-02 H- .3738231 T- 16672.88 E- .5539168 R- 2.675062 U- .3483583 X- 13.24815
X- 4.973996E-02 H- .3702363 T- 16380.32 E- .558844 R- 2.700978 U- .3440458 X- 13.20729
X- 5.133062E-02 H- .3669692 T- 16113.73 E- .5633394 R- 2.725024 U- .3400485 X- 13.16663
X- 5.291453E-02 H- .3639933 T- 15870.82 E- .567441 R- 2.747303 U- .3363372 X- 13.12613
X- 5.449227E-02 H- .3612826 T- 15649.48 E- .5711832 R- 2.767916 U- .3328859 X- 13.0858
X- 5.606437E-02 H- .3588133 T- 15447.77 E- .5745975 R- 2.786965 U- .3296706 X- 13.04561
X- 5.763131E-02 H- .3565638 T- 15263.96 E- .5777127 R- 2.804547 U- .3266696 X- 13.00555
X- 5.919353E-02 H- .3545146 T- 15096.46 E- .5805548 R- 2.820758 U- .3238634 X- 12.96561
X- 6.075145E-02 H- .3526478 T- 14943.82 E- .583148 R- 2.835691 U- .321234 X- 12.92578
X- 6.230543E-02 H- .350947 T- 14804.71 E- .5855139 R- 2.849433 U- .3187653 X- 12.88605
X- 6.38582E-02 H- .3493976 T- 14677.94 E- .5876725 R- 2.862069 U- .3164425 X- 12.84642
X- 6.540293E-02 H- .347986 T- 14562.4 E- .589642 R- 2.873679 U- .3142524 X- 12.80687
X- 6.694704E-02 H- .3467 T- 14457.11 E- .5914389 R- 2.884338 U- .3121826 X- 12.76739
X- 6.848841E-02 H- .3455282 T- 14361.15 E- .5930784 R- 2.89412 U- .3102223 X- 12.72799
X- 7.002729E-02 H- .3444607 T- 14273.69 E- .5945742 R- 2.903089 U- .3083614 X- 12.68864
X- .0715639 H- .3434881 T- 14193.99 E- .595939 R- 2.911309 U- .3065908 X- 12.64936
X- 7.309843E-02 H- .342602 T- 14121.35 E- .5971842 R- 2.918839 U- .3049022 X- 12.61013
X- 7.463107E-02 H- .3417946 T- 14055.14 E- .5983202 R- 2.925734 U- .3032882 X- 12.57095
X- 7.616197E-02 H- .3410589 T- 13994.8 E- .5993568 R- 2.932045 U- .3017418 X- 12.53181
X- .0776913 H- .3403887 T- 13939.81 E- .6003025 R- 2.937818 U- .300257 X- 12.49271
X- 7.921919E-02 H- .339778 T- 13889.68 E- .6011653 R- 2.943098 U- .2988281 X- 12.45365
X- 8.074577E-02 H- .3392215 T- 13844 E- .6019526 R- 2.947927 U- .2974499 X- 12.41462
X- 8.227115E-02 H- .3387144 T- 13802.36 E- .6026708 R- 2.95234 U- .2961178 X- 12.37563
X- 8.379544E-02 H- .3382524 T- 13764.4 E- .6033261 R- 2.956372 U- .2948277 X- 12.33666
X- 8.531874E-02 H- .3378314 T- 13729.81 E- .603924 R- 2.960056 U- .2935757 X- 12.29771
X- 8.684114E-02 H- .3374478 T- 13698.28 E- .6044695 R- 2.963421 U- .2923582 X- 12.25879
X- .0883627 H- .3370983 T- 13669.54 E- .6049673 R- 2.966494 U- .2911721 X- 12.21989
X- 8.988351E-02 H- .3367797 T- 13643.35 E- .6054214 R- 2.9693 U- .2900147 X- 12.18101
X- 9.140364E-02 H- .3364895 T- 13619.47 E- .6058357 R- 2.971861 U- .2888831 X- 12.14215
X- 9.292314E-02 H- .3362249 T- 13597.7 E- .6062137 R- 2.974199 U- .2877751 X- 12.1033
X- 9.444207E-02 H- .3359839 T- 13577.86 E- .6065586 R- 2.976333 U- .2866885 X- 12.06447
X- 9.596048E-02 H- .3357642 T- 13559.77 E- .6068733 R- 2.978281 U- .2856214 X- 12.02565

X-	.0974784	H-	.335564	T-	13543.29	E-	.6071603	R-	2.980058	U--	.2845719	X-	11.98685
X-	.0989959	H-	.3353814	T-	13528.25	E-	.6074223	R-	2.98168	U--	.2835384	X-	11.94805
X-	.100513	H-	.335215	T-	13514.54	E-	.6076613	R-	2.98316	U--	.2825194	X-	11.90927
X-	.1020297	H-	.3350633	T-	13502.04	E-	.6078793	R-	2.984511	U--	.2815137	X-	11.87049
X-	.1035462	H-	.3349249	T-	13490.64	E-	.6080783	R-	2.985744	U--	.2805199	X-	11.83172
X-	.1050623	H-	.3347987	T-	13480.24	E-	.6082598	R-	2.98687	U--	.279537	X-	11.79296
X-	.1065781	H-	.3346834	T-	13470.74	E-	.6084254	R-	2.987899	U--	.2785639	X-	11.75421
X-	.1080937	H-	.3345781	T-	13462.07	E-	.6085765	R-	2.988839	U--	.2775997	X-	11.71546
X-	.1096091	H-	.3344818	T-	13454.15	E-	.6087143	R-	2.989699	U--	.2766434	X-	11.67672
X-	.1111242	H-	.3343937	T-	13446.9	E-	.6088402	R-	2.990487	U--	.2756945	X-	11.63799
X-	.1126392	H-	.3343129	T-	13440.26	E-	.6089549	R-	2.991209	U--	.274752	X-	11.59926
X-	.114154	H-	.3342387	T-	13434.17	E-	.6090596	R-	2.991874	U--	.2738152	X-	11.56053
X-	.1156686	H-	.3341703	T-	13428.57	E-	.6091551	R-	2.992486	U--	.2728837	X-	11.52181
X-	.1171831	H-	.334107	T-	13423.4	E-	.6092423	R-	2.993053	U--	.2719566	X-	11.48309
X-	.1186974	H-	.3340481	T-	13418.62	E-	.6093218	R-	2.993581	U--	.2710335	X-	11.44438
X-	.1202117	H-	.3339929	T-	13414.16	E-	.6093944	R-	2.994076	U--	.2701136	X-	11.40567
X-	.1217257	H-	.3339408	T-	13409.98	E-	.6094606	R-	2.994543	U--	.2691966	X-	11.36696
X-	.1232397	H-	.3338911	T-	13406.02	E-	.609521	R-	2.994989	U--	.2682817	X-	11.32825
X-	.1247536	H-	.3338429	T-	13402.24	E-	.6095761	R-	2.995421	U--	.2673683	X-	11.28955
X-	.1262673	H-	.3337955	T-	13398.56	E-	.6096264	R-	2.995846	U--	.2664559	X-	11.25085
X-	.127781	H-	.3337481	T-	13394.94	E-	.6096722	R-	2.996272	U--	.2655436	X-	11.21215
X-	.1292945	H-	.3336995	T-	13391.3	E-	.6097141	R-	2.996708	U--	.2646306	X-	11.17346
X-	.1308079	H-	.3336486	T-	13387.56	E-	.6097522	R-	2.997165	U--	.2637161	X-	11.13477
X-	.1323212	H-	.3335941	T-	13383.64	E-	.6097871	R-	2.997655	U--	.262799	X-	11.09608
X-	.1338344	H-	.3335345	T-	13379.42	E-	.6098189	R-	2.99819	U--	.2618782	X-	11.0574
X-	.1353475	H-	.3334679	T-	13374.8	E-	.6098478	R-	2.99879	U--	.2609522	X-	11.01871
X-	.1368603	H-	.333392	T-	13369.61	E-	.6098744	R-	2.999472	U--	.2600193	X-	10.98004
X-	.138373	H-	.3333042	T-	13363.7	E-	.6098985	R-	3.000262	U--	.2590776	X-	10.94137
X-	.1398855	H-	.3332013	T-	13356.85	E-	.6099205	R-	3.001189	U--	.2581247	X-	10.9027
X-	.1413977	H-	.3330795	T-	13348.82	E-	.6099406	R-	3.002287	U--	.2571576	X-	10.86404
X-	.1429096	H-	.3329339	T-	13339.3	E-	.6099589	R-	3.003599	U--	.2561729	X-	10.82539
X-	.1444212	H-	.332759	T-	13327.93	E-	.6099756	R-	3.005178	U--	.2551665	X-	10.78674
X-	.1459323	H-	.3325478	T-	13314.26	E-	.609991	R-	3.007087	U--	.2541333	X-	10.74811
X-	.1474428	H-	.3322918	T-	13297.75	E-	.6100048	R-	3.009403	U--	.2530671	X-	10.70949
X-	.1489528	H-	.3319809	T-	13277.75	E-	.6100175	R-	3.012222	U--	.2519605	X-	10.67089
X-	.1504619	H-	.3316026	T-	13253.48	E-	.6100292	R-	3.015658	U--	.2508046	X-	10.63231
X-	.1519701	H-	.3311419	T-	13223.96	E-	.6100397	R-	3.019853	U--	.2495886	X-	10.59375
X-	.1534771	H-	.3305805	T-	13188.02	E-	.6100494	R-	3.024982	U--	.2482993	X-	10.55522
X-	.1549828	H-	.329896	T-	13144.25	E-	.6100582	R-	3.031258	U--	.2469209	X-	10.51673
X-	.1564866	H-	.3290616	T-	13090.92	E-	.6100662	R-	3.038944	U--	.2454342	X-	10.47829
X-	.1579884	H-	.3280446	T-	13025.96	E-	.6100736	R-	3.048366	U--	.2438162	X-	10.43989
X-	.1594876	H-	.3268054	T-	12946.83	E-	.6100802	R-	3.059925	U--	.2420389	X-	10.40157

THIN SOLUTION

X-	.1594876	H-	.3268054	T-	12946.49	E-	.6100802	R-	3.059925	U--	.2420389	X-	10.40157
X-	.1637641	H-	.3267423	T-	12945.38	E-	.6099077	R-	3.060516	U--	.2395478	X-	10.29224
X-	.1680401	H-	.3266787	T-	12944.26	E-	.6097335	R-	3.061112	U--	.2370573	X-	10.18292
X-	.1723157	H-	.3266144	T-	12943.12	E-	.6095576	R-	3.061714	U--	.2345673	X-	10.07361
X-	.1765909	H-	.3265495	T-	12941.98	E-	.6093799	R-	3.062323	U--	.2320777	X-	9.964314
X-	.1808657	H-	.3264839	T-	12940.83	E-	.6092004	R-	3.062938	U--	.2295887	X-	9.855028
X-	.1851401	H-	.3264177	T-	12939.66	E-	.609019	R-	3.063559	U--	.2271002	X-	9.745753
X-	.189414	H-	.3263508	T-	12938.48	E-	.6088358	R-	3.064188	U--	.2246122	X-	9.63649
X-	.1936874	H-	.3262832	T-	12937.29	E-	.6086507	R-	3.064823	U--	.2221247	X-	9.527237
X-	.1979605	H-	.3262148	T-	12936.08	E-	.6084636	R-	3.065465	U--	.2196378	X-	9.417996
X-	.202233	H-	.3261457	T-	12934.86	E-	.6082745	R-	3.066114	U--	.2171513	X-	9.308766
X-	.2065052	H-	.3260759	T-	12933.63	E-	.6080834	R-	3.06677	U--	.2146655	X-	9.199548
X-	.2107768	H-	.3260054	T-	12932.38	E-	.6078902	R-	3.067434	U--	.2121802	X-	9.090343
X-	.215048	H-	.325934	T-	12931.13	E-	.6076949	R-	3.068106	U--	.2096954	X-	8.981148
X-	.2193187	H-	.3258619	T-	12929.85	E-	.6074974	R-	3.068785	U--	.2072112	X-	8.871965
X-	.223589	H-	.3257889	T-	12928.56	E-	.6072977	R-	3.069472	U--	.2047276	X-	8.762795
X-	.2278588	H-	.3257151	T-	12927.26	E-	.6070957	R-	3.070168	U--	.2022445	X-	8.653637

ORIGINAL PAGE IS
OF POOR QUALITY

X- .2321281	H- .3256405	T- 12925.94	E- .6068913	R- 3.070871	U- .1997621	X- 8.544492
X- .2363968	H- .325565	T- 12924.61	E- .6066847	R- 3.071583	U- .1972802	X- 8.435359
X- .2406651	H- .3254886	T- 12923.25	E- .6064755	R- 3.072304	U- .1947989	X- 8.32624
X- .2449329	H- .3254112	T- 12921.89	E- .6062637	R- 3.073035	U- .1923182	X- 8.217133
X- .2492002	H- .3253329	T- 12920.5	E- .6060494	R- 3.073774	U- .1898381	X- 8.108038
X- .253467	H- .3252537	T- 12919.1	E- .6058323	R- 3.074523	U- .1873586	X- 7.998957
X- .2577332	H- .3251734	T- 12917.68	E- .6056128	R- 3.075282	U- .1848798	X- 7.88989
X- .2619989	H- .3250922	T- 12916.24	E- .6053903	R- 3.076051	U- .1824016	X- 7.780836
X- .2662641	H- .3250099	T- 12914.78	E- .6051649	R- 3.07683	U- .179924	X- 7.671795
X- .2705287	H- .3249265	T- 12913.31	E- .6049367	R- 3.077619	U- .1774471	X- 7.562769
X- .2747928	H- .324842	T- 12911.81	E- .6047053	R- 3.07842	U- .1749708	X- 7.453756
X- .2790563	H- .3247563	T- 12910.29	E- .6044708	R- 3.079232	U- .1724951	X- 7.344758
X- .2833193	H- .3246696	T- 12908.75	E- .6042333	R- 3.080055	U- .1700202	X- 7.235775
X- .2875816	H- .3245816	T- 12907.19	E- .6039925	R- 3.080889	U- .1675459	X- 7.126806
X- .2918434	H- .3244924	T- 12905.61	E- .6037483	R- 3.081736	U- .1650723	X- 7.017852
X- .2961046	H- .3244019	T- 12904.01	E- .6035004	R- 3.082597	U- .1625994	X- 6.908913
X- .3003653	H- .32431	T- 12902.38	E- .6032491	R- 3.083469	U- .1601272	X- 6.799989
X- .3046253	H- .3242169	T- 12900.72	E- .602994	R- 3.084355	U- .1576557	X- 6.691081
X- .3088847	H- .3241223	T- 12899.04	E- .6027352	R- 3.085255	U- .155185	X- 6.582188
X- .3131434	H- .3240263	T- 12897.33	E- .6024725	R- 3.086169	U- .152715	X- 6.473312
X- .3174015	H- .3239288	T- 12895.6	E- .6022055	R- 3.087098	U- .1502457	X- 6.364452
X- .321659	H- .3238298	T- 12893.84	E- .6019345	R- 3.088042	U- .1477772	X- 6.255609
X- .3259158	H- .3237293	T- 12892.05	E- .6016592	R- 3.089001	U- .1453095	X- 6.146782
X- .330172	H- .3236271	T- 12890.24	E- .6013793	R- 3.089977	U- .1428426	X- 6.037972
X- .3344274	H- .3235232	T- 12888.39	E- .601095	R- 3.090969	U- .1403764	X- 5.92918
X- .3386822	H- .3234175	T- 12886.51	E- .6008056	R- 3.091979	U- .1379111	X- 5.820405
X- .3429363	H- .32331	T- 12884.59	E- .6005114	R- 3.093007	U- .1354465	X- 5.711648
X- .3471897	H- .3232006	T- 12882.64	E- .6002119	R- 3.094054	U- .1329829	X- 5.602909
X- .3514423	H- .3230893	T- 12880.66	E- .5999072	R- 3.095119	U- .13052	X- 5.494189
X- .3556942	H- .322976	T- 12878.64	E- .5995969	R- 3.096206	U- .1280581	X- 5.385488
X- .3599454	H- .3228605	T- 12876.58	E- .5992808	R- 3.097313	U- .1255969	X- 5.276805
X- .3641958	H- .3227428	T- 12874.48	E- .5989587	R- 3.098442	U- .1231367	X- 5.168143
X- .3684454	H- .3226229	T- 12872.34	E- .5986304	R- 3.099594	U- .1206774	X- 5.059501
X- .3726942	H- .3225006	T- 12870.15	E- .5982955	R- 3.10077	U- .118219	X- 4.95088
X- .3769422	H- .3223758	T- 12867.92	E- .5979539	R- 3.10197	U- .1157616	X- 4.842278
X- .3811893	H- .3222484	T- 12865.65	E- .5976052	R- 3.103196	U- .1133051	X- 4.733698
X- .3854357	H- .3221183	T- 12863.32	E- .5972491	R- 3.104449	U- .1108496	X- 4.62514
X- .3896811	H- .3219854	T- 12860.94	E- .5968852	R- 3.105731	U- .1083951	X- 4.516604
X- .3939257	H- .3218496	T- 12858.51	E- .5965135	R- 3.107041	U- .1059416	X- 4.408091
X- .3981693	H- .3217107	T- 12856.02	E- .5961331	R- 3.108383	U- .1034891	X- 4.2996
X- .4024121	H- .3215685	T- 12853.47	E- .5957438	R- 3.109758	U- .1010377	X- 4.191134
X- .4066539	H- .3214229	T- 12850.87	E- .5953452	R- 3.111166	U- .9.858742E-02	X- 4.082691
X- .4108947	H- .3212737	T- 12848.19	E- .5949368	R- 3.112611	U- .0961382	X- 3.974274
X- .4151345	H- .3211207	T- 12845.44	E- .5945179	R- 3.114094	U- .0936901	X- 3.865881
X- .4193733	H- .3209637	T- 12842.63	E- .594088	R- 3.115618	U- .9.124316E-02	X- 3.757516
X- .423611	H- .3208024	T- 12839.73	E- .5936465	R- 3.117184	U- .8.879738E-02	X- 3.649177
X- .4278477	H- .3206366	T- 12836.75	E- .5931926	R- 3.118796	U- .0863528	X- 3.540865
X- .4320832	H- .320466	T- 12833.68	E- .5927258	R- 3.120456	U- .8.390945E-02	X- 3.432583
X- .4363176	H- .3202904	T- 12830.52	E- .5922448	R- 3.122167	U- .8.146736E-02	X- 3.324329
X- .4405508	H- .3201092	T- 12827.26	E- .5917488	R- 3.123934	U- .7.902654E-02	X- 3.216106
X- .4447829	H- .3199221	T- 12823.89	E- .5912367	R- 3.125761	U- .7.658699E-02	X- 3.107913
X- .4490136	H- .3197285	T- 12820.4	E- .5907068	R- 3.127653	U- .7.414875E-02	X- 2.999753
X- .453243	H- .3195278	T- 12816.78	E- .5901573	R- 3.129618	U- .7.171182E-02	X- 2.891626
X- .4574711	H- .319319	T- 12813.01	E- .5895855	R- 3.131665	U- .6.927611E-02	X- 2.783533
X- .4616978	H- .3191006	T- 12809.06	E- .5889877	R- 3.133808	U- .6.684156E-02	X- 2.675477
X- .465923	H- .3188706	T- 12804.9	E- .5883581	R- 3.136068	U- .0644079	X- 2.567459
X- .4701466	H- .3186257	T- 12800.47	E- .5876877	R- 3.138479	U- .0619748	X- 2.459481
X- .4743685	H- .3183609	T- 12795.68	E- .5869626	R- 3.14109	U- .5.954162E-02	X- 2.351547
X- .4785886	H- .3180681	T- 12790.37	E- .586161	R- 3.143981	U- .5.710737E-02	X- 2.243659
X- .4828066	H- .3177349	T- 12784.32	E- .5852489	R- 3.147278	U- .5.467056E-02	X- 2.135824

X- .4870222 H- .3173423 T- 12777.19 E- .5841743 R- 3.151171 U--5.222899E-02 X- 2.028052
X- .4912349 H- .316862 T- 12768.45 E- .5828592 R- 3.155948 U--4.977947E-02 X- 1.920353
X- .495444 H- .3162521 T- 12757.33 E- .5811897 R- 3.162034 U--4.731777E-02 X- 1.812747
X- .4996484 H- .3154532 T- 12742.73 E- .5790026 R- 3.170042 U--4.483832E-02 X- 1.70526
X- .5038465 H- .3143824 T- 12723.09 E- .5760713 R- 3.180839 U--4.233421E-02 X- 1.597934
X- .5080363 H- .3129279 T- 12696.31 E- .5720894 R- 3.195624 U--3.979735E-02 X- 1.490822
X- .5122145 H- .3109429 T- 12659.53 E- .5666552 R- 3.216025 U--3.721874E-02 X- 1.384003
X- .516377 H- .3082411 T- 12609.06 E- .5592588 R- 3.244214 U--3.458936E-02 X- 1.277589
X- .5205181 H- .304594 T- 12540.16 E- .5492743 R- 3.283059 U--3.190143E-02 X- 1.171721
X- .5246303 H- .299731 T- 12446.91 E- .5359615 R- 3.336325 U--2.914985E-02 X- 1.06659
X- .528704 H- .2933452 T- 12321.97 E- .5184798 R- 3.408953 U--2.633427E-02 X- .9624441
X- .5327272 H- .2851036 T- 12156.41 E- .4959175 R- 3.507497 U--2.346151E-02 X- .85959
X- .5366849 H- .2746641 T- 11939.38 E- .4673384 R- 3.64081 U--2.054768E-02 X- .7584121
X- .5405587 H- .2616991 T- 11657.79 E- .4318454 R- 3.821182 U--1.762001E-02 X- .6593767
X- .5443271 H- .2459231 T- 11295.74 E- .3886571 R- 4.066311 U--1.471805E-02 X- .5630354
X- .5479647 H- .2271244 T- 10833.77 E- .3371938 R- 4.402873 U--1.189385E-02 X- .4700403
X- .5514418 H- .2051959 T- 10247.85 E- .2771623 R- 4.873392 U--9.210446E-03 X- .3811474
X- .5547239 H- .1801624 T- 9507.817 E- .2086305 R- 5.550548 U--6.739009E-03 X- .2972403
X- .5577708 H- .1522 T- 8575.265 E- .1320806 R- 6.570303 U--4.554446E-03 X- .2193447
X- .5605347 H- .1216437 T- 7400.493 E- 4.842985E-02 R- 8.220727 U--2.730065E-03 X- .148685
X- .562955 H- 8.898176E-02 T- 5918.152 E--4.098555E-02 R- 11.23826 U--1.331357E-03 X- 8.68
X- .5649468 H- 5.483396E-02 T- 4040.891 E--.1344686 R- 18.23687 U--4.102183E-04 X- 3.58887
X- .5663506 H- .0199174 T- 1650 E--.2300561 R- 50.20735 U- 0 X- 0

ORIGINAL PAGE IS
OF POOR QUALITY

SUMMARY OUTPUT
Output from file SAMP.HT

The line below "QC = " is radiative heating from the individual bands of the eight step model. The order here corresponds to Olstad's original numbering scheme, which is slightly different than that presented in AIAA Paper 88-2672. I. E.

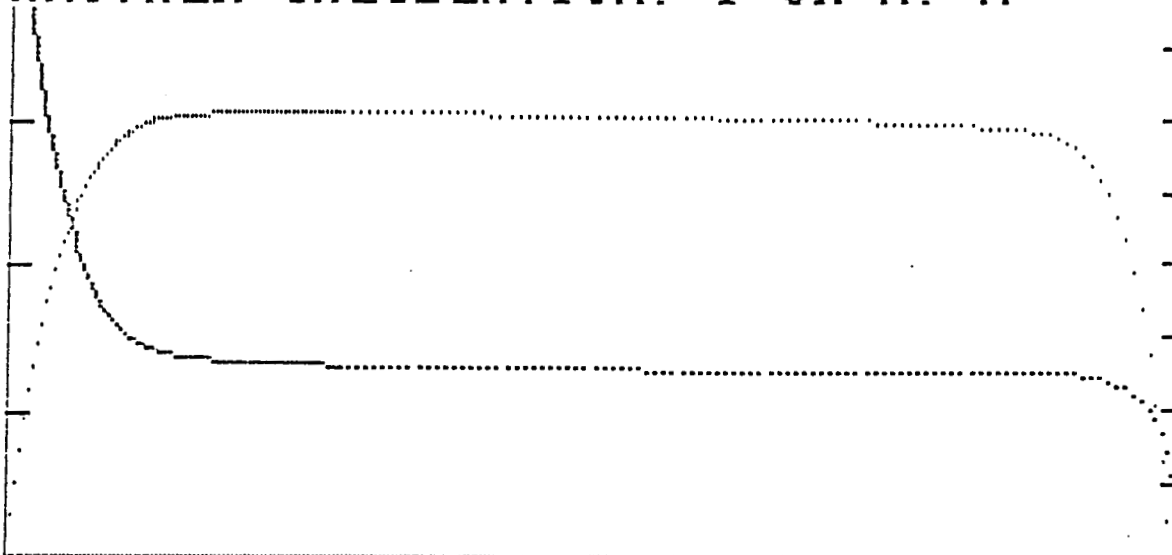
Bands		Wavelength (A)	Includes
AIAA 88-2672	Olstad		
1	1	400-852	VUV Continuum
2	2	852-911	VUV Continuum
3	3	911-1020	VUV Continuum
4	4	1020-1130	VUV Continuum
5	6	1130-1801	Continuum + Line Wings
6	7	1130-1801	Line "Centers"
7	5	1801-4000	"Visible"
8	8	4000-	Visible + Infrared

In the output VUV is bands 1-4, VUV lines is bands 5-6 (6-7 in Olstad's numbers), and VIS+IR is bands 7 and 8 (5 and 8 in Olstad's scheme).

U1= 1600000 P1= 24.897
T2= 13053.98 E2= .6271315
S1= 9.189011E-17 S2= 1.345287E-21
B1= 4.794425E+08 B2= 4.260638E+11
XR= .8430347 L= 25.56528
N= 3 T1= 200.15
TW= 1650 PLOT TMAX= 50000
PLOT EPS MAX= 1 - 0
E2= .6271315 EQ= .6101499
IM= 100 DTAU= .075 XS= 14.47891
RB = 230
QC= 133.1875 WATTS/SQ CM
34.29741 81.80688 291.3677 198.9687 63.1825 296.8814 3.33415 213.7095
VUV 606.4407
VUV LINES 300.2155
VIS+IR 276.8921 QRTOT= 1183.548

SCREEN PLOT
Output of file SAMP.BAS
Retrieval by PLOTPIC.BAS

QC= 133.1875 WATTS/SQ CM
UUU = 606.4407 XS= 14.47891
LINE REGION = 300.2155 RB = 230
VIS AND IR = 276.8921 QRTOT= 1183.548
ANOTHER CALCULATION? Y OR N? N



ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX 6

The AFE Programs

Inverse Programs for Computing Nonequilibrium Inviscid Radiating Flowfields
about Blunt Bodies

Developed by
Leland A. Carlson, Glenn J. Bobskill, and Robert B. Greendyke

USER INSTRUCTIONS AND COMMENTS

The nonequilibrium inviscid inverse codes developed as part of this project are all modifications of a code originally developed by William L. Grose of the NASA Langley Research Center and described in NASA TN D-6529, "A Thin Shock Layer Solution for Nonequilibrium, Inviscid, Hypersonic Flows in Earth, Martian, and Venusian Atmospheres", December 1971. Several different versions of these new codes have been created, and some of these are still in a state of development. In addition, since these codes were developed by several individuals over several years, they are not as general as originally hoped. Whenever possible, the non-general features of the various codes will be pointed out as part of this appendix.

It should be noted that various versions of the codes were developed corresponding to different chemical and electron temperature models. In addition, for each physical model, two different codes were developed. The first, which is termed the mainframe version, was specifically coded for the IBM 3090 and is a completely integrated code with many output files for graphics, etc. The second, called the microcomputer version, was coded for an AT compatible and actually consists of three separate parts. The first part sets up user selected input parameters via keyboard input and reads input files. The second part computes the chemically reacting nonequilibrium flowfield and creates outputfiles necessary for the radiation calculations, and the third part computes the radiative heat transfer from the computed flowfield. The last part can be run separately as long as the input files associated with a given flowfield are available. By making changes in these files of the input parameters controlling various options, the radiative heat transfer associated with various electron temperature and radiation models can be computed for a variety of cases very rapidly. This feature is very advantageous when studying various radiation and electron temperature models. The microcomputer version could easily be used on a mainframe machine.

In general, the input data, variable names, etc. are those used originally in NASA TN D-6529; and the user is referred to that document for details. As in the original NASA version, the shock shape is hard coded in the program. In the present versions, the shock shape corresponds approximately to that for a 60 degree hyperboloid or sphere cone body having a nose radius of about 230 cm. For different body shapes, the shock shape equation would have to be changed. Further, since the method is an inverse thin shock layer method, its computed body shape will be "inaccurate" in the immediate vicinity of the stagnation point; and for very blunt bodies the sonic line far away from the stagnation point will not be accurately located. However, for the regions discussed in the publications associated with this project, the method should be very accurate.

As indicated, the present codes are modifications of the original Grose code. While they retain the ability to handle up to 25 species and 25 forward-reverse reactions (counted as 50 separate reactions by the codes), and reaction rates and the reactions themselves can be modified via the input data, some changes have been introduced. Normally, the present codes assume the use of 10 species and 11 forward-backward reactions and the ordering of these species and reactions should be considered fixed without extensive and careful checking of the code.

For example, species number 1 must be N_2 and species 8 must be N_2^+ . Also, if included reaction 10 ($j=19$ and 20 in the code) must be the nitrogen electron impact ionization reaction $N + e = N^+ + 2e$ in all versions. In some versions of the code, reaction 11 ($j=21$ and 22) must be the corresponding oxygen electron impact ionization reaction. Note that the ordering of the reactions in the code is not that which was given in AIAA Paper 88-2673. The actual reactions used in any version of the code can be easily determined by looking at the corresponding input data.

One of the modifications included in all versions of the code is the ability to compute either an entire flowfield or a "partial flowfield". For the entire flowfield situation, both the mainframe and microcomputer versions normally use twenty streamlines. However, in some cases, in order to resolve the flowfield in more detail and permit more accurate radiation calculations, additional streamlines are needed. Thus, the mainframe version can also handle forty streamlines. Because of some internal coding, the number of streamlines must either be 20 or 40. This variable is the first variable on the second line of the detailed input file.

It should be noted that one of the input variables is NSR, the number of streamlines in the "stagnation region", defined as the region between $X=0$ and $X=DELX$ along the shock wave. As described in NASA TN D-6529, Grose normally used $NSR=5$. However, for the present studies it was found that $NSR=3$ was adequate. Since there is a possibility of some internal hard coding depending upon this value, it should be kept at a value of three without careful checking of the code. With this value, streamline number four is computed twice. The first corresponds to the body, while the second is for the streamline crossing the shock wave at $DELX$. After that, streamlines are computed starting at $DELX$ and then proceeding in $DELX$ increments along the shock wave.

For the "partial flowfield" option, the user can select six streamlines to be actually computed. Of these the first two must be less than $NSR+1$ and the remaining four larger. The partial flowfield option is very useful for making rapid nonequilibrium chemistry, electron temperature, and vibration dissociation coupling studies. However, since it does not compute a detailed flowfield, its results cannot be used to compute radiative heat transfer.

Another modification included in all versions of the codes is the option to select from several vibration dissociation coupling models. The options available are vibrational equilibrium, CVD coupling, CVDV coupling, CVDV preferential coupling, and the new MCVDV model. Further details concerning these models and examples of their similarities and differences are presented in AIAA Paper 88-2673 and in the detailed results contained in Appendix 3. For all the nonequilibrium options, the user also has the choice to select either Millikan and White vibrational relaxation data for N_2 and N_2^+ or the data included in the input files. For unknown reasons, the Millikan and White correlations were hard coded into the program, which is why N_2 must be species number 1 and N_2^+ must be species eight. It is recommended that the Millikan and White data always be used. Further, based upon the studies of this project, the MCVDV and CVDV models are probably the most robust and "realistic". It should be noted that the CVDV preferential model, while theoretically sound, is computationally intensive.

The various codes also contain options concerning the shock jump conditions. The rationale, possible usage, and some results for these various options are discussed in the thesis by Robert Greendyke; but, in general, these options have not been thoroughly tested. Thus, it is suggested that only the frozen flow shock jump option (IQ3 = 1) be utilized without further investigation.

Another major modification to the original Grose code is the inclusion of radiation models and the computation of radiative heat transfer to the vehicle. These calculations are performed after the chemically reacting blunt body flowfield has been obtained. However, the flowfield portion has been modified to trap and appropriately store flowfield properties at the locations where the streamlines cross the X grid lines. Then in the radiation portion of the new codes, the tangent slab approximation is utilized to compute the radiation to the vehicle. In the mainframe version, the user has the option to select various radiative heat transfer (absorption coefficient) models for the non-gray absorbing-emitting computation of the heating. These models are described in AIAA Paper 88-2673 and the thesis of Robert Greendyke, and it is believed that the best of these models is the eight step absorption coefficient model. Consequently, because of memory limitations, the microcomputer version only includes the eight step model. Further, several of the models contain a modified eight step model slightly different from that used in AIAA 88-2673. These later modifications were introduced in order to yield better agreement with the detailed radiation model RADICAL (NASA CR-1656).

It should be noted that in the programs and their output that the numbering of the bands or steps is different for the eight step model than that presented in AIAA 88-2673. The relationship is as follows:

AIAA 88-2673 Band	Program Band	Wavelength (Å)
1	1	400-852
2	2	852-911
3	3	911-1020
4	4	1020-1130
5	6	1130-1801
6	7	1130-1801
7	5	1801-4000
8	8	4000-

The radiation portion of the new programs also contains various options to handle nonequilibrium effects on radiative heating. Basically, nonequilibrium effects are accounted for by "correction factors" which are computed as functions of temperature and composition. Corrections on atomic radiation are controlled by the input variable IQ5 (Yes = 0, No = 1) and those on molecular radiation are selected via IQ8 (Yes = 0, No = 1). In addition, in some versions of the codes, the user can also select whether the molecular corrections should be computed using translational temperature (IQ6=1) or electron temperature (IQ6=0). Note that in early versions of the codes (AFE2 and AFEM) molecular corrections cannot be selected without also selecting atomic corrections. In the later versions which

compute electron temperatures independently from vibrational temperatures, (AFETE and AFENEW), all possible combinations of IQ5 and IQ8 values are permissible with the eight step radiation model. (The other models in the mainframe version are still limited in that molecular corrections can only be included if atomic corrections are selected.)

There is also an IQ9 = 0 option which gives details of the radiation calculation at each point in the program for each wavelength step. This option is only used occassionally. Also, in the mainframe versions the IS7 parameter identifies which streamline for which all flowfield values are stored for subsequent graphical output of the variation of properties along the streamline. In general, in the mainframe version the body and shock coordinates are also stored along with those for the selected streamline. Programs to produce graphical output from these files have been written and utilize the TAMU version of the DISSPLA system. While not included in this report, copies will be provided the contractor upon request.

VERSIONS

As previously mentioned, in general there are for each flowfield model two versions of the code -- one corresponding to a mainframe version (IBM 3090) and one suitable for a micro or minicomputer (VAX or AT class). The latter version could easily be adapted for rapid computations on a mainframe machine.

"Original Version" and Modified Version

This version of the code is the closest of the existing codes to that which was used for the results presented in AIAA Paper 88-2673. It assumes that the electron temperature is equal to the nitrogen vibrational temperature in the radiation calculations and in all chemical rates governed by electron temperature. It also assumes that the atomic oxygen ionization reaction involves a general collision partner, M. I.E. $O + M = O^+ + e + M$. In the output for this code, at each step where the composition and flow properties are printed, values are also printed for two electron temperatures, TE1 and TE2. These were "estimates" obtained from some correlations previously developed from experimental data at lower velocities and temperatures. However, for the conditions of interest in the present study, they appear to be inappropriate and should be ignored.

The microcomputer version consists of a series of programs which may either be executed via a batch file or run independently sequentially. The first of these is INPUT. This program asks the user for a series of inputs which are associated with the various program options and the freestream conditions. This information is stored in a file called GEN.INPUT.

The second program in the sequence is AFE2.FOR, which reads GEN.INPUT and an input data file RR3.NUM. The last name is arbitrary and is supplied by the user during the execution of INPUT. AFE2.FOR computes the chemically reacting nonequilibrium flowfield and subsequently creates three files, RADIN1.DAT, RADIN2.DAT, and RADIN3.DAT, which provide the data needed for the radiation calculation by the third program, called RAD.FOR. This latter program computes

the nongray emitting absorbing radiative flux to the wall using the original eight step absorption coefficient model described in AIAA 88-2673.

The modified version is essentially the same except that the radiation program is replaced by the program RADMOD.FOR. As previously discussed, this modified version has the absorption coefficients for VUV lines in steps 6 and 7 (program numbering scheme) reduced so as to yield better agreement with RADICAL. The changes are only in those statements which compute $KAPPA(K,6,J)$ and $KAPPA(K,7,J)$.

In this report, only the listing for the mainframe version of the modified code is included, and it is called AFEM. This program would have to be compiled and executed with appropriate job control language. The latter must set up input output units having numbers 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 69, 70, 71, 72, 73, 74, and 75. The values above 15 correspond to files associated with storing data for later retrieval by graphics programs. The other files are described on the first page of the listing. The input to the program is provided by two files. The first, called SCREEN, contains the user selected options and the freestream conditions; and the second, identical to RR3.NUM, contains the remaining required input data. Note that in the listing of SCREEN contained in this report, there is a variable IQ10. This value is not used by AFEM and should be deleted when SCREEN is used as input for AFEM.

"Three" Temperature Version

This version has essentially been developed since the end of the project, but it is included in this report as a matter of completeness. Instead of assuming that the electron temperature is equal to the nitrogen vibrational temperature, this model computes a separate free electron temperature using a quasi-equilibrium form of the electron energy equation. In addition, it assumes that atomic ionization due to electron impact occurs as a two step process; and, thus, it uses modified reaction chemistry and reaction rates based upon the experimental work of Wilson. Also, the nonequilibrium radiation correction factors are computed in this version by a simpler, although equivalent, scheme than that used in AFE2 and AFEM. (This new approach is based upon a suggestion by Robert Greendyke and eliminates the need to compute the equivalent equilibrium composition. The results are the same, however, as the previous model.) This model is still under development and further changes may occur in the future. In general, this model is probably most applicable at higher speeds ($U > 11.5$ km/sec) where significant numbers of free electrons are present. Results obtained with this model and the theory behind its development will be presented in a paper at the June 1989 AIAA Thermophysics Conference.

The microcomputer code version of this model utilizes the programs INPUT.FOR, AFETE.FOR, and RADNW1.FOR with input files RR3W.NUM; and the mainframe version uses AFENEW with input files SCREEN and KDRW. In SCREEN the value of IQ10 determines which values are used in the radiation calculation. If $IQ10=1$, the electron temperature is determined by the quasi-equilibrium approach. If $IQ10=0$, the electron temperature is assumed to be equal to the nitrogen vibrational temperature. In both cases, the electron impact ionization reactions are assumed to be controlled by an electron temperature determined by

the quasi-equilibrium approach for free electron temperature. It should be noted that in AFETE.FOR that IQ10 is hard coded to be unity. This value, of course, could be easily changed. In addition, since the value of IQ10 is written to the file RADIN1.DAT by AFETE.FOR and since the radiation program RADNW1.FOR can be run independently, the value of IQ10 can be changed to the desired value in the RADIN1 file by the user prior to executing RADNW1.

Further, AFENEW and RADNW1 both allow all possible combinations of options on usage of the radiation correction factors in the eight step model as determined by the values of IQ5 and IQ8. Also, in the output at each printed flowfield point in AFENEW and AFETE, the value printed for TE1 is the electron temperature determined by the quasi-equilibrium approach. On the same line, is printed a scaled value of the local electron production rate, WE, due to all reactions.

STATEMENT LISTING FOR PROGRAM INPUT.FOR

```

C
C PROGRAM WHICH SUPPLIES INPUT DATA FOR PROGRAM AFE2.FOR
C THROUGH GEN.INPUT
C
REAL RINPUT(6)
INTEGER IINPUT(15),IS(6)
CHARACTER*30 FILES(4)
C
WRITE(*,1)
1 FORMAT(1X,'PRINT FREQUENCY - ', $)
READ(*,*) IINPUT(1)
WRITE(*,2)
2 FORMAT(1X,'DO YOU WANT TO INPUT AN INITIAL INTEGRATION STEP?',
*/,1X,'Yes = 0 , No = 1 ( 1 = Def. Value of 1.220703125E-04 ) : '
*, $)
READ(*,*) IAXI
IF (IAXI.EQ.1) THEN
  RINPUT(1) = .0625/512.0
  GO TO 4
ENDIF
WRITE(*,3)
3 FORMAT(1X,'INITIAL INTEGRATION STEP ( XI.LE.0.5 ) - ', $)
READ(*,*) RINPUT(1)
4 WRITE(*,6)
6 FORMAT(1X,'VIBRATIONAL EQUILIBRIUM ? Yes = 0 , No = 1 : ', $)
READ(*,*) IINPUT(2)
WRITE(*,7)
7 FORMAT(1X,'ENTER DELX ALONG SHOCK (cm) - ', $)
READ(*,*) RINPUT(2)
WRITE(*,8)
8 FORMAT(1X,'NO. OF PSI LINES IN THE STAG. REGION (MAX=10) - ', $)
READ(*,*) IINPUT(3)
NSR1 = IINPUT(3) + 1
WRITE(*,9)
9 FORMAT(1X,'LENGTH OF SYMMETRY AXIS , Z , (cm) - ', $)
READ(*,*) RINPUT(3)
WRITE(*,10)
10 FORMAT(1X,'DO YOU WANT THE FULL FLOW FIELD SOLUTION ?',/,
* 25X,'Yes = 0 , No = 1 : ', $)
READ(*,*) IINPUT(4)
IF (IINPUT(4).EQ.0) GO TO 12
WRITE(*,11) NSR1
11 FORMAT(1X,'ENTER STRMLINE NO. 1,2,3,4,5,6 FOR INFO. EXTRACTION'
*/,1X,' 1 < NO.1 < NO.2 < ' ,I2,' .LE. NO.3 < NO.4 < NO.5 < NO.6 '
*/,1X,' - ', $)
READ(*,*) (IS(I),I=1,6)
C
12 IF (IINPUT(2).EQ.0) GO TO 15
WRITE(*,13)
13 FORMAT(1X,'DO YOU WANT MILLIKAN AND WHITE DATA (N2, N2+) ?',
*/,1X,'Yes = 1, No = 2 - ', $)
READ(*,*) IINPUT(5)
WRITE(*,14)
14 FORMAT(1X,'VIBRATION-DISSOCIATION COUPLING MODELS :',/,
* 1X,'TYPE : 1 FOR CVD MODEL',/,10X,'2',2X,
* 'FOR CVDV MODEL ( excluding NO and NO+ ) ',/,10X,'3',
* 2X,'FOR CVDV-Pref. ( excluding NO and NO+ ) ',/,
* 10X,'4',2X,'FOR PARK MODEL ( with M&W selected ) - ', $)
READ(*,*) IINPUT(6)
15 WRITE(*,16)

```

ORIGINAL PAGE IS
OF POOR QUALITY


```

16  FORMAT(1X,'SHOCK JUMP MODELS : ',/,
*    1X,'TYPE :   1  FOR FROZEN FLOW ACROSS THE SHOCK',/,8X,
*    '   2  FOR N2 FROZEN AND O2 DISSOCIATING',/,8X,
*    '   3  FOR N2 & O2 DISSOCIATING          - ',$,)
    READ(*,*) IINPUT(7)
    IF (IINPUT(7).EQ.2.OR.IINPUT(7).EQ.3) THEN
        WRITE(*,17)
17  FORMAT(1X,'Tvib - Tinf (1) OR Tvib - Ttrans (2) - ',$,)
        READ(*,*) IINPUT(8)
        WRITE(*,18)
18  FORMAT(1X,' Tel - Tinf (1) OR Tel - Ttrans (2) - ',$,)
        READ(*,*) IINPUT(9)
    ENDIF
    WRITE(*,19)
19  FORMAT(1X,'RADIATION MODELS : ',/,1X,
*    'TYPE :   0  NO MODEL ',/,1X,
*    '   1  OLSTAD MODEL ',/,1X,
*    '   2  CARLSON MODEL',/,1X,
*    '   3  BOTH MODELS',/,1X,
*    '   4  BOTH MODELS WITH CAPTURED VALUES          - ',$,)
    READ(*,*) IINPUT(10)
C
    IF (IINPUT(10).NE.0) THEN
        WRITE(*,20)
20  FORMAT(1X,'QR VALUES AT EACH STEP (Yes - 0, No - 1) - ',$,)
        READ(*,*) IINPUT(15)
        WRITE(*,21)
21  FORMAT(1X,'NON-EQUIL. CORR. FOR ATOMIC RAD. (Y=0,N=1) - ',$,)
        READ(*,*) IINPUT(11)
        WRITE(*,22)
22  FORMAT(1X,'NON-EQUIL. CORR. FOR MOLECULAR RAD. (Y=0,N=1) - ',$,)
        READ(*,*) IINPUT(14)
        IF (IINPUT(14).EQ.0) THEN
            WRITE(*,23)
23  FORMAT(1X,'BETA AS A FUNCTION OF (Te - 0, Tt - 1) - ',$,)
            READ(*,*) IINPUT(13)
        ENDIF
    ENDIF
    WRITE(*,24)
24  FORMAT(1X,'ENTER STRLNE NO. FOR DETAILED CALC. (None - 0) - ',$,)
    READ(*,*) IINPUT(12)
C
    WRITE(*,25)
25  FORMAT(1X,'FREESTREAM VELOCITY (cm/sec)          - ',$,)
    READ(*,*) RINPUT(4)
    WRITE(*,26)
26  FORMAT(1X,'FREESTREAM PRESSURE (dynes/cm**2)      - ',$,)
    READ(*,*) RINPUT(5)
    WRITE(*,27)
27  FORMAT(1X,'FREESTREAM TEMPERATURE ( K )          - ',$,)
    READ(*,*) RINPUT(6)
C
    WRITE(*,30)
30  FORMAT(1X,'ENTER NAME OF INPUT DATA FILE        : ',$,)
    READ(*,39) FILES(1)
    WRITE(*,32)
32  FORMAT(1X,'ENTER NAME OF OUTPUT DATA FILE       : ',$,)
    READ(*,39) FILES(2)
    WRITE(*,34)
34  FORMAT(1X,'ENTER NAME OF EVIS OUTPUT FILE        : ',$,)

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

      READ(*,39) FILES(3)
      IF (IINPUT(10).GT.0) WRITE(*,35)
35  FORMAT(1X,'ENTER NAME OF RADIATION OUTPUT FILE : ', $)
      IF (IINPUT(10).GT.0) READ(*,39) FILES(4)
39  FORMAT(A30)
C
      OPEN (UNIT=106,FILE='GEN.INPUT',STATUS='UNKNOWN')
      WRITE(106,*) (IINPUT(I),I=1,15)
      WRITE(106,*) (IS(I),I=1,6)
      WRITE(106,*) (RINPUT(I),I=1,6)
      IF (IINPUT(10).EQ.0) WRITE(106,70) (FILES(I),I=1,3)
      IF (IINPUT(10).GT.0) WRITE(106,75) (FILES(I),I=1,4)
70  FORMAT(1X,3(A30,2X))
75  FORMAT(1X,4(A30,2X))
      CLOSE(106)
      END

```

ORIGINAL PAGE IS
OF POOR QUALITY

STATEMENT LISTING FOR PROGRAM AFE2.FOR

PROGRAM AFE2

ORIGINAL PAGE IS
OF POOR QUALITY

AN INVERSE HYPERSONIC FLOW SOLUTION FOR AN AFE/AOTV BODY IN
CHEMICAL NONEQUILIBRIUM USING VIBRATION-DISSOCIATION COUPLING MODELS,
SHOCK JUMP APPROXIMATIONS, ELECTRON TEMPERATURE MODELING, AND
UNCOUPLED RADIANT HEAT TRANSFER USING A METHOD DERIVED BY S.MASLEN

BASED ON AN ORIGINAL PROGRAM BY B.L. WEIGEL FOR WILLIAM L. GROSE
VIRGINIA POLYTECHNIC INSTITUTE - 1969, (NASA TN D-6529, DEC. 1971)

PROGRAM MODIFICATION BY:

TEXAS A&M UNIVERSITY
DEPARTMENT OF AEROSPACE ENGINEERING
TAMRF-5671 (1987-1988)

PRINCIPAL INVESTAGATOR :	DR. LELAND A. CARLSON
RESEARCH ASSISTANT :	GLENN J. BOBSKILL
RESEARCH ASSISTANT :	ROBERT B. GREENDYKE

CALINTH : INTEGRATION ROUTINE BY CHARLES E. TREANOR
THE NUMERICAL INTEGRATION ALGORITHM USED IS FOUND IN A METHOD
FOR THE NUMERICAL INTEGRATION OF COUPLED FIRST ORDER
DIFFERENTIAL EQUATIONS WITH GREATLY DIFFERENT TIME CONSTANTS

FOFE - EVALUATE E BY NEWTON ITERATION METHOD
FOFTS1 - EVALUATE TS BY NEWTON ITERATION METHOD, CHEMISTRY FROZEN
FOFTS23 - FOFTS1 WITH N2 FROZEN, O2 DISSOCIATING OR BOTH N2 & O2 DISS.
FTLUP - INTERPOLATION ROUTINE

DATA FILES

UNIT 7 IS USED FOR THE INPUT DATA FILE
UNIT 8 IS USED FOR THE OUTPUT DATA FILE
UNIT 9 IS USED TO STORE VIBRATIONAL ENERGY DATA AT THE SHOCK (FORMATTED)
UNIT 10 IS USED TO STORE ALL STAGNATION QUANTITIES
UNIT 11 IS USED TO STORE ALL EVIS DATA FOR COMPUTATIONAL PURPOSES
UNIT 12 IS USED TO STORE ALL PRESSURE DATA
UNIT 13 IS USED TO STORE ALL DPDX DATA
UNIT 14 IS USED TO STORE QUANTITIES FOR PHYSICAL SPACE CALCULATIONS
UNIT 15 IS USED TO STORE QUANTITIES FOR RAD.FOR
UNIT 16 IS USED TO STORE QUANTITIES FOR RAD.FOR
UNIT 17 IS USED TO STORE QUANTITIES FOR RAD.FOR

INPUT-NAMELIST

DELX - INCREMENT ALONG SHOCK , cm
ZSTERM - LENGTH OF SYMMETRY AXIS , Z , cm
IMAX - MAX. NO. OF I-S SPECIES , LESS THAN OR EQUAL TO 25
JMAX - MAX. NO. OF J-S REACTIONS , LESS THAN OR EQUAL TO 50
MJ - CODE INDICATING WHICH SPECIES , I , TO USE TO CALCULATE
COUPLING FACTOR, PHI SUB J , FOR REACTION J
M - 1 FOR VIBRATIONAL NON-EQUILIBRIUM
- 0 FOR VIBRATIONAL EQUILIBRIUM
R - UNIVERSAL GAS CONSTANT , erg/(mole-K)
GAMMA - RATIO OF SPECIFIC HEATS
CIINF - FREESTREAM MASS FRACTION FOR EACH SPECIES

C PINF - FREESTREAM PRESSURE , dynes/cm**2
C TINF - FREESTREAM TEMPERATURE , K
C VINP - FREESTREAM VELOCITY , cm/sec
C MUI - MOLECULAR WT. FOR EACH SPECIES , gm/mole
C THETAI - CHARACTERISIC VIBRATIONAL TEMPERATURE , K
C DGENI - FUDGE FACTOR TO PERMIT APPROXIMATING POLYATOMIC MOLECULE
C BY A DIATOMIC MOLECULE
C FI - 0 FOR MONATOMIC SPECIES
C - 1 FOR ALL OTHERS
C DELHI - HEAT OF FORMATION , ergs/mole
C DELI - DISSOCIATION ENERGY OF SPECIES , K
C EVI - VIBRATIONAL ENERGY OF SPECIES , ergs/g
C BI - $3/2 \ln(2 \pi M_i k / h^2) + \ln(k) + FI(I) \ln(THETAI(Rotational))$
C + $\ln(GIL(I,1))$ FOR EACH SPECIE I
C LI - NUMBER OF ELECTRONIC LEVELS FOR EACH SPECIES (LI.LE.20)
C GIL - DEGENERACY OF L-TH ELECTRONIC LEVEL FOR I-TH SPECIES
C EPSIIL - L-TH ELECTRONIC ENERGY LEVEL FOR I-TH SPECIES , K
C AJ - FREQUENCY FACTOR IN ARRENIUS TYPE RATE EQN.
C BJ - TEMPERATURE EXPONENT IN ARRENIUS TYPE EQN.
C EJ - ACTIVATION ENERGY IN ARRENIUS TYPE EQN.
C DIRECT - DIRECTION OF THE REACTION (FORW. - 1.0 , BACK. - 2.0)
C AIJ - FACTOR TO ALLOW USE OF GENERAL SPECIES IN REACTION EQNS.
C i.e. (N2 + M = 2N + M), AIJ = 1.0 OR (N2 + O2 = 2NO), AIJ = 0.0
C NUIJ - STOICHIOMETRIC COEFFICIENTS OF I-TH REACTANT IN J-TH REACTION
C NUPIJ - STOICHIOMETRIC COEFFICIENTS FOR I-TH PRODUCT IN J-TH REACTION
C ALPIK - FACTORS IN EQN. FOR VIBRATIONAL RELAXATION TIME
C BETAIK - FACTORS IN EQN. FOR VIBRATIONAL RELAXATION TIME
C SIGIK - FACTORS IN EQN. FOR VIBRATIONAL RELAXATION TIME
C NIP - NUMBER OF VIBRATIONAL LEVELS FOR ANHARMONIC OSCILLATOR , K
C UP - CHARACTERISTIC PROBABILITY TEMPERATURE , K
C WE - CONST. USED TO COMPUTE ANHARMONIC OSCILLATOR , K
C WEXE - CONST. USED TO COMPUTE ANHARMONIC OSCILLATOR , K
C WEYE - CONST. USED TO COMPUTE ANHARMONIC OSCILLATOR , K
C WEZE - CONST. USED TO COMPUTE ANHARMONIC OSCILLATOR , K
C XI - INITIAL COMPUTING INTERVAL , .0001220703125 UNLESS INPUT , cm
C ELE1 - (2*IMAX + 1) VALUES USED BY INTEGRATION SCHEME
C NORMALLY 0.1,0.5, OR .05 , (UPPER PH RANGE)
C ELE2 - (2*IMAX + 1) VALUES USED BY INTEGRATION SCHEME
C NORMALLY .05,.1, OR .01 AND .LT. ELE1 , (LOWER PH RANGE)
C XPST - 99 OR LESS Xs AT WHICH PHYSICAL SPACE CALCULATIONS ARE
C DESIRED. THEY MUST BE MULTIPLES OF DELX IN ORDER TO HAVE
C RS,COST,ZS, AND SINT VALUES AND LAST MUST BE .GT. X AT
C ZSTERM. THEREFORE, XPST(NXPST) SET = X AT ZSTERM + 100.0
C IN PROGRAM. XPST(1) MAY NOT BE 0.0 . THEREFORE, SET
C XPST(1) = DELX IN PROGRAM
C NXPST - NUMBER OF Xs AT WHICH PHYSICAL SPACE CALCULATIONS ARE
C DESIRED
C CIMAX - MAX. CJ OR COMPUTING INTERVAL 0.0625 UNLESS INPUT OTHERWISE
C HCHCKT - CONTROL ON SIZE OF COMPUTING INTERVAL IN CHECK
C IF(ABS(HPREV - H)/H.GT.HCHECK) REDUCE INTERVAL
C TCHCKT - CONTROL ON SIZE OF COMPUTING INTERVAL IN CHECK
C IF(ABS(TPREV - T)/T.GT.TCHECK) REDUCE INTERVAL
C PHMAX - CONTROL ON COMPUTING INTERVAL .LE. 65.0
C IPF - OUTPUT PRINT FREQUENCY
C
C STOPS
C -----
C STOP 1 INCORRECT INPUT
C STOP 2 IN MAIN WHEN NXPST IS .LE. 2
C STOP 13 IN SHOCKG

```

C STOP 30 IN CHECK WHEN COMPUTING INTERVAL .LT. 1.0E-15
C STOP 66 IN BASIC WHEN NO CONVERGENCE ON E ITERATION (FOFE)
C STOP 301 IN MAIN FOR ERROR IN XPST ARRAY OR IZTERM .GT. 200
C STOP 321 IN MAIN WHEN NO CONVERGENCE ON TS ITERATION (FOFTS)
C STOP 663 IN MAIN WHEN X .NE. VARI(IPSI)
C STOP 665 IN MAIN AFTER AN INTEGRATION ATTEMPT
C STOP 670 IN MAIN WHEN A CI NEGATIVE
C
C NOTE : PROGRAM IS CURRENTLY CONFIGURED FOR A MAXIMUM
C 200 STREAMLINE SOLUTION
C
C IMPLICIT REAL*8(A-H,O-Z)
COMMON /A1/ M,IX,IMAX,IPSI,MODEL,TE1,TE2
COMMON /A2/ P(200),DPDX(200),VARI(200)
C FOLLOWING 7 VARIABLES DIMENSIONED BY IMAX
COMMON /A3/ EVIINF(25),THETAI(25),MUI(25),FI(25),DELHI(25),
* CIINF(25),LI(25)
COMMON /A4/ R,MUINF,DELTA,LAMSQ,OMEGSQ,OMEGA,MU,T,U,TINF
C FOLLOWING 2 VARIABLES DIMENSIONED BY (IMAX IN LI,IMAX)
COMMON /A5/ EPSIIL(20,25),GIL(20,25),CID(200,4)
COMMON /A6/ VAR(52),CUVAR(52),DER(51)
C FOLLOWING 4 VARIABLES DIMENSIONED BY JMAX
COMMON /A7/ MJ(50),EJ(50),AJ(50),BJ(50),DIRECT(50)
C FOLLOWING 3 VARIABLES DIMENSIONED BY IMAX
COMMON /A8/ TVI(25),NI(25),DGENI(25),BI(25)
C EVIS(IMAX)
COMMON /A9/ EVIS(25),XPST(100)
C FOLLOWING 3 VARIABLES DIMENSIONED BY (IMAX+1,JMAX) OR (IMAX,JMAX)
COMMON /A10/ NUIJ(26,50),AIJ(25,50),NUPIJ(25,50)
C FOLLOWING 3 VARIABLES DIMENSIONED BY (IMAX,JMAX)
COMMON /A11/ SIGIK(25,25),ALPIK(25,25),BETAIK(25,25)
COMMON /A12/ SINTM(200),COSTM(200),RSM(200),RCM(200),X1(200)
* ,ZSM(200)
COMMON /A13/ DELX,ZSTERM,IZTERM,NSR,MW,SP,TS
COMMON /A14/ EINF,PINF,RHOINF,VINF,E,JMAX,KEYINT,RHO,HSTAG
COMMON /A15/ PFTL,KITR1,NIP(25),UP(25)
COMMON /A16/ IUNEG,WE(25),WEZE(25),WEYE(25),WEZE(25)
COMMON /A17/ ELB,SPEC,CJ,TPREV,HPREV,HCHECK,TCHECK
COMMON /A18/ ITNEG,IEXP
COMMON /A19/ ELE1(51),ELE2(51),NERR
C DELI(IMAX)
DIMENSION DELI(25)
DIMENSION RUT(200),PSISTG(200)
DIMENSION CIG(25,11),EIG(25,11)
DIMENSION PSIG(11),TG(11),HG(11),CIT(11),EIT(11),EG(11)
C CM(IMAX),CI(IMAX),EVI(IMAX)
DIMENSION CM(25),CI(25),EVI(25)
EQUIVALENCE (VAR(1),XVAR),(VAR(2),H),(VAR(3),CI(1))
REAL*8 MUI,MUINF,MU,MINF,MINFSQ,LAMBDA,LAMSQ
CHARACTER*30 INFILE,OUTFL1,OUTFL2,OUTFL3
CHARACTER*30 SPECIE(25),REACT(50)
C OPEN GENERAL INPUT FILE CREATED BY INPUT.FOR
OPEN (UNIT=106,FILE='GEN.INPUT',STATUS='OLD')
C READ INTERACTIVE INPUTS FROM GEN.INPUT
READ(106,*) IPF,M,NSR,IAN,MW,MODEL,IQ3,IQ1,IQ2,IQ4,
* IQ5,IQ7,IQ6,IQ8,IQ9
READ(106,*) IS1,IS2,IS3,IS4,IS5,IS6
READ(106,*) XI,DELX,ZSTERM,VINF,PINF,TINF
IF (IQ4.EQ.0) READ(106,20) INFILE,OUTFL1,OUTFL2
IF (IQ4.GT.0) READ(106,18) INFILE,OUTFL1,OUTFL2,OUTFL3

```

```

18  FORMAT(1X,4(A30,2X))
20  FORMAT(1X,3(A30,2X))
    CLOSE(106)
    NSR1 = NSR + 1
    NERR = 0
    ITNEG = 0
    IEXP = 0
    NEG = 0
    IUNEG = 0
    DO 21 I = 1,52
    VAR(I) = 0.0
    CUVAR(I) = 0.0
    IF (I.EQ.52) GO TO 21
    DER(I) = 0.0
    ELE1(I) = 0.0
    ELE2(I) = 0.0
21  CONTINUE
    OPEN (UNIT=7,FILE=INFILE,STATUS='OLD')
    OPEN (UNIT=8,FILE=OUTFL1,STATUS='UNKNOWN')
    OPEN (UNIT=15,FILE='RADIN1.DAT',STATUS='UNKNOWN')

```

C

READ INPUT

```

    READ(7,*) IMAX,JMAX,R,GAMMA
    READ(7,*) NXPST,CIMAX,PHMAX,TCHCKT,HCHCKT
    NXCON = NXPST - 1
    WRITE(15,40) OUTFL3
40  FORMAT(A30)
    WRITE(15,*) MODEL,IQ3,VINF,PINF,TINF
    WRITE(15,*) NXPST,NXCON,IMAX,IQ4,IQ5,IQ6,IQ8,IQ9
    DO 42 I = 1,IMAX
    READ(7,*) THETAI(I),MUI(I),FI(I),DELHI(I),CIINF(I),LI(I),DGENI(I)
    *      ,DELI(I),EVI(I),BI(I)
42  WRITE(15,*) MUI(I)
    DO 45 J = 1,JMAX,2
45  READ(7,*) MJ(J),AJ(J),BJ(J),EJ(J),DIRECT(J)
    DO 48 J = 2,JMAX,2
48  READ(7,*) MJ(J),DIRECT(J)
    IMAX21 = 2*IMAX + 1
    DO 51 I = 1,IMAX21
51  READ(7,*) ELE1(I),ELE2(I)
    DO 54 I = 1,IMAX
54  READ(7,*) (AIJ(I,J),J = 1,JMAX)
    DO 57 J = 1,JMAX
57  READ(7,*) (NUIJ(I,J),I = 1,IMAX+1)
    DO 60 J = 1,JMAX
60  READ(7,*) (NUPIJ(I,J),I = 1,IMAX)
    DO 63 I = 1,IMAX
63  READ(7,*) (ALPIK(J,I),J = 1,IMAX)
    DO 66 I = 1,IMAX
66  READ(7,*) (BETAIK(J,I),J = 1,IMAX)
    DO 69 I = 1,IMAX
69  READ(7,*) (SIGIK(J,I),J = 1,IMAX)
    DO 72 I = 1,IMAX
    LII = LI(I)
    WRITE(15,*) LII
    READ(7,*) (GIL(L,I),L = 1,LII)
72  WRITE(15,*) (GIL(L,I),L = 1,LII)
    DO 75 I = 1,IMAX
    LII = LI(I)
    WRITE(15,*) LII
    READ(7,*) (EPSIIL(L,I),L = 1,LII)

```

```

75 WRITE(15,*) (EPSIIL(L,I),L = 1,LII)
   IF (IANS.EQ.1) THEN
     NXPST = 4
     XPST(1) = (IS3 - NSR)*DELX
     XPST(2) = (IS4 - NSR)*DELX
     XPST(3) = (IS5 - NSR)*DELX
     XPST(4) = (IS6 - NSR)*DELX
     GO TO 78
   ENDIF
   DO 76 I = 1,NXPST
     XPST(I) = DELX*I
76  WRITE(15,*) I,XPST(I)
78  DO 80 I = 1,IMAX
80  READ(7,*) NIP(I),UP(I),WE(I),WEXE(I),WEYE(I),WEZE(I)
     IMAXP2 = IMAX + 2
C                                     SUM OF CIINF SHOULD BE 1.0
     SUM = 0.0
     DO 81 I = 1,IMAX
81  SUM = SUM + CIINF(I)
     IF (SUM.EQ.1.0) GO TO 85
     WRITE(*,82) SUM
82  FORMAT(1X,'SUM OF CIINF(I) SHOULD BE 1: IT IS - ',E11.4,' STOP 1')
     STOP
C                                     STOP 1
85  HCHECK = HCHCKT
     TCHECK = TCHCKT
C                                     READ IN AND WRITE OUT SPECIES AND REACTIONS
     DO 88 I = 1,IMAX
88  READ(7,90) SPECIE(I)
90  FORMAT(A30)
     WRITE(8,95)
     WRITE(*,95)
95  FORMAT(//)
     WRITE(8,100)
     WRITE(*,100)
100 FORMAT(14X,'SPECIES'//)
     DO 110 I = 1,IMAX
105 FORMAT(1X,I5,11X,A30)
     WRITE(*,105) I,SPECIE(I)
110 WRITE(8,105) I,SPECIE(I)
     WRITE(*,95)
     WRITE(8,95)
     DO 112 I = 1,JMAX
112 READ(7,90) REACT(I)
     WRITE(*,115)
     WRITE(8,115)
115 FORMAT(14X,'REACTIONS'//)
     DO 120 I = 1,JMAX
     WRITE(*,125) I,REACT(I)
120 WRITE(8,125) I,REACT(I)
125 FORMAT(1X,I5,8X,A30)
     WRITE(*,95)
     WRITE(8,95)
     CLOSE(7)
C                                     PRINT INPUT
     WRITE(8,135) HCHCKT,TCHCKT,PHMAX
     WRITE(*,135) HCHCKT,TCHCKT,PHMAX
135 FORMAT(1X,'HCHCKT = ',E11.4,4X,'TCHCKT = ',E11.4,4X,'PHMAX = ',
*      E11.4/)
     WRITE(8,140) XI,CIMAX,NXPST

```



```

WRITE(*,140) XI,CIMAX,NXPST
140 FORMAT(1X,'XI = ',E11.4,8X,'CIMAX = ',E11.4,5X,'NXPST = ',I3//)
DO 141 I = 1,IMAX21
WRITE(*,142) I,ELE1(I),I,ELE2(I)
141 WRITE(8,142) I,ELE1(I),I,ELE2(I)
142 FORMAT(1X,'ELE1(',I2,') = ',E11.4,5X,'ELE2(',I2,') = ',E11.4)
WRITE(8,95)
WRITE(*,95)
DO 143 I = 1,NXPST
WRITE(*,144) I,XPST(I)
143 WRITE(8,144) I,XPST(I)
144 FORMAT(1X,'XPST(',I3,') = ',E11.4)
WRITE(8,95)
WRITE(*,95)
WRITE(8,145) IMAX,JMAX,M
WRITE(*,145) IMAX,JMAX,M
145 FORMAT(1X,'IMAX = ',I2,11X,'JMAX = ',I2,13X,'M = ',I2//)
WRITE(8,150) DELX,ZSTERM,IPF,R,PINF,TINF,VINF,GAMMA
WRITE(*,150) DELX,ZSTERM,IPF,R,PINF,TINF,VINF,GAMMA
150 FORMAT(1X,'DELX = ',E11.4,2X,'ZSTERM = ',E11.4,2X,
*'PRINT FREQ. = ',I3,3X,'R = ',E11.4,/,1X,'PINF = ',E11.4,
*2X,'TINF = ',E11.4,4X,'VINF = ',E11.4,2X,'GAMMA = ',E9.3//)
WRITE(8,155)
WRITE(*,155)
155 FORMAT(16X,'MUI',8X,'THETAI',7X,'DGENI',10X,'FI',9X,'DELHI'//)
DO 157 I = 1,IMAX
WRITE(*,160) I,MUI(I),THETAI(I),DGENI(I),FI(I),DELHI(I)
157 WRITE(8,160) I,MUI(I),THETAI(I),DGENI(I),FI(I),DELHI(I)
160 FORMAT(1X,I2,8X,5(E11.4,2X))
WRITE(8,95)
WRITE(8,161)
WRITE(*,95)
WRITE(*,161)
161 FORMAT(16X,'DELI',8X,'CIINF',9X,'EVI',10X,'LI',11X,'BI'//)
DO 163 I = 1,IMAX
WRITE(*,164) I,DELI(I),CIINF(I),EVI(I),LI(I),BI(I)
163 WRITE(8,164) I,DELI(I),CIINF(I),EVI(I),LI(I),BI(I)
164 FORMAT(1X,I2,8X,3(E11.4,2X),3X,I3,7X,E11.4)
WRITE(8,95)
WRITE(8,165)
WRITE(*,95)
WRITE(*,165)
165 FORMAT(8X,'NIP',9X,'UP',10X,'WE',10X,'WEXE',9X,'WEYE',8X,'WEZE'//)
DO 167 I = 1,IMAX
166 FORMAT(1X,I2,5X,I3,4X,E10.4,2X,E10.4,3X,E10.4,3X,E10.4,2X,E10.4)
WRITE(*,166) I,NIP(I),UP(I),WE(I),WEXE(I),WEYE(I),WEZE(I)
167 WRITE(8,166) I,NIP(I),UP(I),WE(I),WEXE(I),WEYE(I),WEZE(I)
WRITE(8,95)
WRITE(8,170)
WRITE(*,95)
WRITE(*,170)
170 FORMAT(12X,'MJ',9X,'AJ',14X,'BJ',15X,'EJ',10X,'DIRECTION',/,
* 69X,'F-1,B-2'//)
DO 180 J = 1,JMAX,2
175 FORMAT(1X,I2,2X,I8,3(2X,E15.6),5X,F5.2)
WRITE(*,175) J,MJ(J),AJ(J),BJ(J),EJ(J),DIRECT(J)
180 WRITE(8,175) J,MJ(J),AJ(J),BJ(J),EJ(J),DIRECT(J)
WRITE(8,95)
WRITE(8,181)
WRITE(*,95)

```

```

      WRITE(*,181)
181  FORMAT(12X,'MJ',6X,'DIRECTION',/,21X,'F-1,B-2'/)
      DO 183 J = 2,JMAX,2
182  FORMAT(1X,I2,2X,I8,8X,F5.2)
      WRITE(*,182) J,MJ(J),DIRECT(J)
183  WRITE(8,182) J,MJ(J),DIRECT(J)
      WRITE(8,95)
      WRITE(*,95)
      DO 187 I = 1,IMAX
      LII = LI(I)
      DO 186 L = 1,LII
      WRITE(8,185) L,I,GIL(L,I),L,I,EPSIIL(L,I)
      WRITE(*,185) L,I,GIL(L,I),L,I,EPSIIL(L,I)
185  FORMAT(1X,'GIL(',I2,',',I2,',') = ',E11.4,4X,
      *      'EPSIIL(',I2,',',I2,',') = ',E11.4)
186  CONTINUE
187  CONTINUE
      WRITE(8,95)
      WRITE(*,95)
      DO 200 I = 1,IMAX
      DO 196 J = 1,JMAX
      WRITE(*,195) I,J,AIJ(I,J)
      WRITE(8,195) I,J,AIJ(I,J)
195  FORMAT(1X,'AIJ(',I2,',',I2,',') = ',E11.4)
196  CONTINUE
200  CONTINUE
      WRITE(8,95)
      WRITE(*,95)
      DO 205 I = 1,IMAX+1
      DO 202 J = 1,JMAX
      WRITE(8,201) I,J,NUIJ(I,J),I,J,NUPIJ(I,J)
      WRITE(*,201) I,J,NUIJ(I,J),I,J,NUPIJ(I,J)
201  FORMAT(1X,'NUIJ(',I2,',',I2,',') = ',I5,9X,'NUPIJ(',I2,',',
      *      'I2,',') = ',I5)
202  CONTINUE
205  CONTINUE
      WRITE(8,95)
      WRITE(*,95)
      DO 225 I = 1,IMAX
      DO 220 J = 1,IMAX
      WRITE(8,210) I,J,SIGIK(I,J),I,J,ALPIK(I,J),I,J,BETAIK(I,J)
      WRITE(*,210) I,J,SIGIK(I,J),I,J,ALPIK(I,J),I,J,BETAIK(I,J)
210  FORMAT(1X,'SIGIK(',I2,',',I2,',') = ',E10.4,1X,
      *      'ALPIK(',I2,',',I2,',') = ',E10.4,1X,'BETAIK(',I2,',',I2,',') = ',
      *      'E10.4)
220  CONTINUE
225  CONTINUE
      WRITE(8,95)
      WRITE(8,230)
      WRITE(*,95)
      WRITE(*,230)
230  FORMAT(7X,25('*'),' END INPUT ',25('*')///)
      WRITE(8,231)
      WRITE(*,231)
231  FORMAT(25X,'+++ SHOCK GEOMETRY +++'///)
      NXPSTM1 = NXPST - 1
C
C                                     DEFINE SHOCK GEOMETRY
      CALL SHOCKG (NXPSTM1,ITK)
C
C                                     IZTERM = NO. OF DELTA X INCREMENTS GENERATED IN SHOCKG
      IF (IQ4.GT.0.AND.IZTERM.GT.30) THEN

```

```

WRITE(*,232) IZTERM
232 FORMAT(1X,'IZTERM = ',I4,2X,'WITH A RADIATION MODEL SELECTED'
*      ,/,1X,'MAXIMUM NO. OF STREAMLINES .LE. 30')
      STOP
      ENDIF
      WRITE(8,95)
      WRITE(*,95)
      IF ((NXPST - 1.0).LE.IZTERM) GO TO 237
      WRITE(*,235)
235 FORMAT(1X,'(NXPST-1).GT.IZTERM : IT MUST BE .LE. TO IZTERM',/,
*      1X,'PHYSICAL SPACE CALCULATIONS - SEE DO 700 LOOP : STOP 301'/)
      STOP
237 WRITE(8,95)
      WRITE(*,95)
      IF (IZTERM.LE.200) GO TO 241
      WRITE(*,240)
240 FORMAT(1X,'IZTERM.GT.200:CHGE DIM OF P,DPDX AND VARI: STOP 301'/)
      STOP

C                                     FREESTREAM QUANTITIES
241 SUM = 0.0
      DO 245 I = 1,IMAX
245 SUM = SUM + CIINF(I)/MUI(I)
      MUINF = 1.0/SUM
      RHOINF = MUINF*PINF/(R*TINF)
      AINF = DSQRT(GAMMA*R*TINF/MUINF)
      MINF = VINP/AINF

C                                     FOR EACH I , SPECIE
      EINF = 0.0
      DO 275 I = 1,IMAX
      TEM = DEXP(THETAI(I)/TINF)
      EVIINF(I) = (R*THETAI(I))/(MUI(I)*(TEM - 1.0))*FI(I)

C                                     FOR EACH L, REACTION LEVEL
      GSUM = 0.0
      GESUM = 0.0
      LII = LI(I)
      IF (LII.LE.20) GO TO 255
      WRITE(*,250) LII
250 FORMAT(1X,'LII = ',I3,2X,'A LEVEL IN LI ARRAY IS GREATER THAN 20',
*      2X,/,1X,'YOU NEED TO CHANGE DIMENSION OF EPSIIL AND GIL')
255 CONTINUE
      DO 265 L = 1,LII
      TEM1 = DEXP(-EPSIIL(L,I)/TINF)
      GSUM = GSUM + GIL(L,I)*TEM1
      GESUM = GESUM + GIL(L,I)*EPSIIL(L,I)*TEM1
265 CONTINUE
      EEIINF = R/MUI(I)*(GESUM/GSUM)
      EIINF = 1.5*R*TINF/MUI(I) + FI(I)*R*TINF/MUI(I) + EVIINF(I)
*      + EEIINF + DELHI(I)/MUI(I)
      EINF = EINF + EIINF*CIINF(I)
275 CONTINUE
      WRITE(8,280) MUINF,RHOINF,AINF,TINF,PINF,VINF,MINF,EIINF,EINF,
*      EEIINF
      WRITE(*,280) MUINF,RHOINF,AINF,TINF,PINF,VINF,MINF,EIINF,EINF,
*      EEIINF
280 FORMAT(24X,'+++ FREESTREAM QUANTITIES +++',///,6X,'MUINF = ',E11.4
*      ,3X,'RHOINF = ',E11.4,3X,'AINF = ',E11.4,///,6X,'TINF = ',E11.4,3X,
*      'PINF = ',E11.4,3X,'VINF = ',E11.4,///,1X,'MINF = ',E10.4,2X,
*      'EIINF = ',E10.4,2X,'EINF = ',E10.4,2X,'EEIINF = ',E10.4//)
      DO 285 I = 1,IMAX
      WRITE(*,286) I,EVIINF(I)

```

```

285 WRITE(8,286) I,EVIINF(I)
286 FORMAT(1X,'EVIINF(',I2,') - ',E11.4)
WRITE(8,95)
WRITE(*,95)

C BEGIN QUANTITIES BEHIND SHOCK FOR THE RANGE OF X
WRITE(8,290)
WRITE(*,290)
290 FORMAT(22X,'+++ QUANTITIES BEHIND SHOCK +++',///,5X,'TS',9X,'ES',
*9X,'PS',7X,'RHOS',8X,'US',8X,'PSIS',8X,'HS'/)
IF (NXPST.GE.3) GO TO 293
WRITE(*,291)
STOP
291 FORMAT(1X,'NXPST = 2 - SURELY SOME PHYSICAL SPACE VALUES',/,1X,
* 'ARE DESIRED -- MAKE IT AT LEAST 3 : STOP 2')
293 OPEN (UNIT=9,FILE=OUTFL2,STATUS='UNKNOWN')
OPEN (UNIT=10,FILE='STAG.DAT',STATUS='UNKNOWN')
OPEN (UNIT=11,FILE='EVISC.DAT',STATUS='UNKNOWN')

C
DO 385 IX = 1,IZTERM
TEM = SINTM(IX)**2
LAMBDA = RHOINF*VINP*SINTM(IX)
OMEGA = PINF + RHOINF*VINP**2*TEM
DELTA = EINF + PINF/RHOINF + (VINP**2*TEM)/2.0
LAMSQ = LAMBDA**2
OMEGSQ = OMEGA**2
MINFSQ = MINF**2
TEM = TEM*MINFSQ
TSG = (TINF*(2.0*GAMMA*TEM - (GAMMA - 1.0))*((GAMMA - 1.0)
* *TEM + 2.0))/((GAMMA + 1.0)**2*TEM)
TOL1 = .001
TOL2 = .0001
MAXI = 50

C
IF (IQ3.EQ.1) CALL FOFTS1 (MAXI,TOL1,TOL2,ICODE,TSG)
IF (IQ3.GT.1) CALL FOFTS23 (MAXI,TOL1,TOL2,ICODE,TSG,IQ1,IQ2,IQ3)

C
IF (ICODE.EQ.1) WRITE(*,330) TSG
330 FORMAT(1X,'**** MAXIMUM ITERATION EXCEEDED IN FOFTS ****',/,
* 1X,'LAST ITERATED VALUE OF TSG WAS ',E15.6,' : STOP 321'/)
IF (ICODE.EQ.2) WRITE(*,335)
335 FORMAT(1X,'***** DERIVATIVE = 0.0 IN FOFTS ***** : STOP 321'/)
IF (ICODE.EQ.1.OR.ICODE.EQ.2) STOP

C STOP 321
IF (IX.EQ.1) WRITE(15,*) TSG
TS = TSG
ES = 0.0
DO 360 I = 1,IMAX
IF (M.EQ.1) EVIS(I) = EVIINF(I)
IF (M.EQ.1) GO TO 345
TEM = DEXP(THETAI(I)/TS)
EVIS(I) = (R*THETAI(I))/(MUI(I)*(TEM - 1.0))*FI(I)
345 SUMG = 0.0
SUMGE = 0.0
LII = LI(I)
DO 350 L = 1,LII
TEM1 = DEXP(-EPSIIL(L,I)/TS)
SUMG = SUMG + TEM1*GIL(L,I)
350 SUMGE = SUMGE + TEM1*GIL(L,I)*EPSIIL(L,I)
EEIS = (R/MUI(I))*(SUMGE/SUMG)
EIS = (1.5*R*TS)/MUI(I) + (FI(I)*R*TS)/MUI(I) + EVIS(I)

```

```

*      + EEIS + DELHI(I)/MUI(I)
  IF (IQ3.EQ.1) THEN
    ES = ES + EIS*CIINF(I)
  ELSE
    ES = ES + EIS*CID(IX,I)
  ENDIF
360  CONTINUE
  PS = DSQRT(OMEGSQ - 2.0*LAMSQ*(DELTA - ES))
  RHOS = (MUINF*PS)/(R*TS)
  US = VINFCOSTM(IX)
  PSIS = (RHOINF*VINFCRSM(IX)**2)/2.0
  HS = PS/RHOS + ES
  WRITE(10,*) IX,TS,ES,PS,RHOS,US,PSIS,HS
  WRITE(8,365) TS,ES,PS,RHOS,US,PSIS,HS
  WRITE(*,365) TS,ES,PS,RHOS,US,PSIS,HS
365  FORMAT(1X,7(E9.4,2X))
  WRITE(9,366) IX
366  FORMAT(1X,'FOR IZTERM = ',I3//)
  DO 370 I = 1,IMAX
    WRITE(9,367) I,EVIS(I)
367  FORMAT(13X,'EVIS(',I2,') = ',E11.4)
370  CONTINUE
  WRITE(11,*) (EVIS(I),I = 1,IMAX)
  WRITE(9,620)
385  CONTINUE
C
  CLOSE(9)
  CLOSE(10)
  CLOSE(11)
  OPEN (UNIT=10,FILE='STAG.DAT',STATUS='OLD')
  OPEN (UNIT=11,FILE='EVIS.DAT',STATUS='OLD')
C
  COMPUTE PRESSURE DISTRIBUTION FOR EACH PSI
  OPEN (UNIT=12,FILE='PRESS.DAT',STATUS='UNKNOWN')
  DO 390 IPSI = 1,IZTERM
    DO 388 IX = 1,IZTERM
      IF (IX.EQ.IPSI) XX = X1(IX)
      READ(10,*) IXT,TS,ES,PS,RHOS,US,PSIS,HS
      IF (IXT.LT.IPSI) GO TO 388
      IF (IXT.EQ.IPSI) PSISHK = PSIS
      IF (IPSI.EQ.IX) THEN
        P(IX) = PS
        GO TO 388
      ENDIF
      P(IX) = PS + US/(RCM(IX)*RSM(IX))*(PSISHK - PSIS)
388  CONTINUE
      CLOSE(10)
      OPEN (UNIT=10,FILE='STAG.DAT',STATUS='OLD')
      WRITE(12,*) XX,(P(IX),IX = 1,IZTERM),IPSI,IZTERM
390  CONTINUE
      CLOSE(12)
      OPEN (UNIT=12,FILE='PRESS.DAT',STATUS='OLD')
      OPEN (UNIT=13,FILE='DPDX.DAT',STATUS='UNKNOWN')
C
  COMPUTE DPDX AND VARI
  DO 404 IPSI = 1,IZTERM
    READ(12,*) X,(P(I),I=1,IZTERM),IPI,IZTERM
    ITHIS = 0
    DO 401 IX = IPSI,IZTERM
      IF (IX.EQ.IPSI) XT = X
      IF (IX.GT.IPSI.AND.IPSI.LT.(NSR+2)) XT = XT + DELX/NSR
      IF (IX.GT.IPSI.AND.IPSI.GT.(NSR+1)) XT = XT + DELX

```

```

      ITHIS = ITHIS + 1
      VARI(ITHIS) = XT
401  CONTINUE
      DO 403 IX = IPSI, IZTERM
C                                     DIFFERENTIATE
      IF (IX.NE.IPSI) GO TO 402
      DPDX(IX) = (- 3.*P(IX) + 4.*P(IX+1) - P(IX+2))/(2.0*DELX)
      GO TO 403
402  DPDX(IX) = (- P(IX-1) + P(IX+1))/(2.0*DELX)
      IF (IX.NE.IZTERM) GO TO 403
      DPDX(IX) = (P(IX-2) - 4.*P(IX-1) + 3.*P(IX))/(2.0*DELX)
403  CONTINUE
      WRITE(13,*) (DPDX(I), I=1, IZTERM), IPSI, IZTERM
404  CONTINUE
      CLOSE(13)
      CLOSE(12)
      OPEN (UNIT=12, FILE='PRESS.DAT', STATUS='OLD')
      OPEN (UNIT=13, FILE='DPDX.DAT', STATUS='OLD')
C
      DO 405 IX = 1, IZTERM
      IF (IX.EQ.1) VARI(1) = 0.0
      IF (IX.EQ.1) GO TO 405
      IF (IX.LT.(NSR+2)) VARI(IX) = VARI(IX-1) + DELX/NSR
      IF (IX.GT.(NSR+1)) VARI(IX) = VARI(IX-1) + DELX
405  CONTINUE
      WRITE(8,95)
      WRITE(*,95)
      WRITE(8,407) ITK, NXPST
      WRITE(*,407) ITK, NXPST
407  FORMAT(1X, 'XXXXX ITK = ', I3, 5X, 'NXPST = ', I3, ' XXXXX'//)
      IF (ITK.GE.(2*NXPST/4)) GO TO 409
      WRITE(*,408)
408  FORMAT(1X, 'NO. OF X-S IN XPST ARRAY, ITK, IS .LT. .5*NXPST', /
*      , 1X, 'REEXAMINE DELX, NXPST AND XPST ARRAY : STOP 301'//)
      STOP
C                                     STOP 301
C
C  END COMPUTATION OF QUANTITIES BEHIND SHOCK ; INTEGRATE ALONG EACH PSI
C
C  VAR(1) = XVAR
C  VAR(2) = H, ENTHALPY          VAR(2) MUST BE H, WHICH MAY BE + OR -
C
C                                     CONCENTRATION OF SPECIE
C  VAR(3)      = CI(1)
C  VAR(2+IMAX) = CI(IMAX)
C
C                                     EQUILIBRIUM VIBRATIONAL ENERGY
C  VAR(2+IMAX+1) = EVI(1)
C  VAR(2+IMAX+IMAX) = EVI(IMAX)
C
409  WRITE(8,411)
      WRITE(*,411)
411  FORMAT(1X, 'NO. OF VIB. LEVELS TO DISSOCIATION', /)
      DO 410 I = 1, IMAX
C                                     TRUNCATE
      YIY = FI(I)*(DELI(I)/THETAI(I) + 1.0)
      NI(I) = AINT(YIY)
      WRITE(*,412) I, NI(I)
410  WRITE(8,412) I, NI(I)
412  FORMAT(11X, 'NI(', I2, ') = ', I3)
      WRITE(*,95)

```

```

WRITE(8,95)
MU = 0.0
N = 1 + 2*IMAX
IF (M.EQ.0) N = N - IMAX

C                                     INITIALIZE

WRITE(8,415) IZTERM
WRITE(*,415) IZTERM
415 FORMAT(1X,'IZTERM = ',I4///)
WRITE(8,417)
WRITE(*,417)
417 FORMAT(1X,'COUPLED VIBRATION-DISSOCIATION MODELS',//,12X,
* 'TYPE',11X,'NO.',//,9X,'VIB. EQUIL.',8X,'0',/,12X,'CVD',13X,
* '1',/,12X,'CVDV',12X,'2',/,6X,'CVDV-Preferential',5X,'3',/,
* 12X,'PARK',12X,'4'///)
WRITE(8,418)
WRITE(*,418)
418 FORMAT(7X,'SHOCK JUMP CONDITION MODELS',//,12X,
* 'TYPE',11X,'NO.',//,6X,'CHEMISTRY FROZEN',6X,'1',/,5X,
*'N2 FROZEN, O2 DISS.',4X,'2',/,7X,'N2 AND O2 DISS.',6X,'3'///)
IF (MW.EQ.1) THEN
WRITE(*,419)
WRITE(8,419)
419 FORMAT(3X,'MILLIKAN AND WHITE DATA FOR N2 WAS SELECTED'/)
ENDIF
KSTAG = 0
ISTAG = 0
XPST(1) = DELX
XPST(NXPST) = VARI(IZTERM) + 100.0
SUMCI = 0.0
WRITE(*,95)
WRITE(8,95)
LPS1 = 1
IPSI = 0
OPEN (UNIT=14,FILE='RHORT.DAT',STATUS='UNKNOWN')
OPEN (UNIT=16,FILE='RADIN2.DAT',STATUS='UNKNOWN')

C                                     BEGIN EACH STREAMLINE COMPUTATION HERE
420 IPSI = IPSI + 1
IF (IPSI.EQ.IZTERM) GO TO 735
425 KIPF = 0

C                                     ISTAG = 2 FOR STREAMLINE DELX
IF (ISTAG.EQ.1) ISTAG = 2
READ(10,*) IX,TS,ES,PS,RHOS,US,PSIS,HS
READ(11,*) (EVIS(I),I = 1,IMAX)
IF (IPSI.EQ.1) HSTAG = HS
CJ = XI
SPEC = 0.0

C                                     EVALUATE DERIVATIVES WHEN SPEC = 0.0
II = 0

C                                     USE SHOCK VALUES FOR INITIAL COMPUTATION ON EACH STREAMLINE
KEYINT = 0
IX = 0
T = TS
IF (M.EQ.0) GO TO 450
DO 445 I = 1,IMAX
445 TVI(I) = TINF
GO TO 460
450 DO 455 I = 1,IMAX
455 TVI(I) = TS
460 RHO = RHOS
IF (IQ3.GT.1) THEN

```

```

      SUM = 0.0
      DO 462 I = 1,IMAX
462    SUM = SUM + CID(IPSI,I)/MUI(I)
      MU = 1.0/SUM
      ENDIF
      IF (IQ3.EQ.1) MU = MUINF
      U = US
      DO 465 I = 1,IMAX
      MM = 2 + I
      K = IMAX + MM
      VAR(MM) = 0.0
      IF (IQ3.EQ.1) VAR(MM) = CIINF(I)
      IF (IQ3.GT.1) VAR(MM) = CID(IPSI,I)
      VAR(K) = 0.0
      IF (M.EQ.0) VAR(K) = EVIS(I)
      IF (M.EQ.1) VAR(K) = EVIINF(I)
      EVI(I) = VAR(K)
465    CONTINUE
C                                     COMPUTE S EXPONENT FOR PARK MODEL
      IF (MODEL.EQ.4) THEN
      SP = 3.5*DEXP(-5000.0/TS)
      ENDIF
      E = ES
      H = HS
      IF ((IPSI.EQ.(NSR+1)).AND.(ISTAG.EQ.0)) GO TO 635
      READ(12,*) X,(P(L),L=1,IZTERM),IPSI,IZTERM
      READ(13,*) (DPDX(L),L=1,IZTERM),IPSI,IZTERM
      VAR(1) = X
C                                     OMIT PSI = 0.0 STREAMLINE FOR NOW - PICK IT UP LATER
      IF (X.NE.0.0) GO TO 490
      IG = 0
      WRITE(*,725)
      WRITE(8,725)
      GO TO 730
490 IF (IANS.EQ.1) THEN
      IF ((IPSI.EQ.IS1).OR.(IPSI.EQ.IS2).OR.(IPSI.EQ.IS3)) GO TO 492
      IF ((IPSI.EQ.IS4).OR.(IPSI.EQ.IS5).OR.(IPSI.EQ.IS6)) GO TO 492
      GO TO 730
      ENDIF
492 IF (DABS(X - VARI(IPSI)).LE.1.0E-06) GO TO 500
      WRITE(*,495) IPSI,X,VARI(IPSI)
495 FORMAT(1X,'IPSI = ',I3,2X,'X = ',E10.5,2X,'VARI(IPSI) = ',E10.5,
      *//,1X,'X AND VARI(IPSI) SHOULD BE EQUAL : EXAMINE GENERATION OF X
      * AND VARI IN MAIN : STOP 663'//)
      STOP
C                                     STOP 663
C                                     INITIALIZE
500 IF (IANS.EQ.1) THEN
      IF ((IPSI.EQ.IS1).OR.(IPSI.EQ.IS2).OR.(IPSI.EQ.IS3)) GO TO 501
      IF ((IPSI.EQ.IS4).OR.(IPSI.EQ.IS5).OR.(IPSI.EQ.IS6)) GO TO 501
      GO TO 730
      ENDIF
501 ELB = 0.0
      SPEC = 0.0
      IF (VAR(1).GT.XPST(LPS1)) LPS1 = LPS1 + 1
      LPS2 = LPS1
505 IX = IX + 1
C                                     CALINTH - MODIFIED RUNGE-KUTTA
      CALL CALINTH (N,CIMAX,PHMAX)
C

```



```

      IF (NERR.EQ.0) GO TO 575
      IF (NERR.EQ.1) WRITE(*,565) CJ,N
565  FORMAT(1X,'BAD INPUT IN CALINTH',/,1X,'CJ = ',E10.5,2X,'N = ',I4
      *      ,/,1X,'STOP 665'/)
      IF (NERR.EQ.2) WRITE(*,570)
570  FORMAT(1X,'EXAMINE ELE1 AND ELE2 IN CALINTH : STOP 665'/)
      IF ((NERR.EQ.1).OR.(NERR.EQ.2)) STOP

```

C STOP 665

```

575  IF (IX.EQ.1) THEN
      IF (IQ3.NE.1) THEN
          IF (IQ2.EQ.1) TE = TINF
          IF (IQ2.EQ.2) TE = TS
          TE1 = TE
          TE2 = TE
      ELSE
          TE = TINF
          TE1 = TE
          TE2 = TE
      ENDIF
  ENDIF

```

C ELECTRON TEMPERATURE MODEL

```

      IF (IX.EQ.1) GO TO 576
      CORR = 4.23E-06*(T**(-2.88))
      CN = RHO*6.02252E+23
      CNN = CN*VAR(5)/MUI(3)
      CNO = CN*VAR(6)/MUI(4)
      TE = DLOG(1. + 10.1/((CNO+CNN)*CORR))
      TE = T/(1. + TE*T/85000.)
      TE1 = TE
      CN2 = RHO*6.02252E+23*VAR(3)/MUI(1)
      A = DLOG(1.01D+01) - DLOG(CN2) - DLOG(6.4D+00)
      *   - 59.*DLOG(1.0D+01) + 19.*DLOG(TVI(1))
      A = DEXP(A)
      TE = DLOG(1. + A)
      TE = TVI(1)/(1. + TE*TVI(1)/85000.)
      TE2 = TE

```

C

```

576  IF (IX.EQ.1) GO TO 580
      IF ((IPSI.EQ.(NSR+1)).AND.(IX.EQ.(NSR+1))) GO TO 580
      IF (VAR(1).GE.VARI(IZTERM)) GO TO 580
      IF ((IPSI.LT.(NSR+1)).AND.(VAR(1).GE.DELX)) GO TO 580
      KIPF = KIPF + 1
      IF (KIPF.NE.IPF) GO TO 625
      KIPF = 0
580  SUMCI = 0.0
      DO 590 ISUM = 3,IMAXP2
          IF (VAR(ISUM).LT.0.0) NEG = -1
          IF (VAR(ISUM).LT.0.0) WRITE(*,585)
585  FORMAT(5X,'----- NEGATIVE CI -----'/)
590  SUMCI = SUMCI + VAR(ISUM)
      DO 595 ICM = 1,IMAX
595  CM(ICM) = VAR(ICM+2)*MU/MUI(ICM)

```

C PRINT ANSWERS

```

      WRITE(8,600) IPSI,IX,IQ3,MODEL,XVAR,H,MU,PFTL,RHO,U,T,E,SPEC,SUMCI
      WRITE(*,600) IPSI,IX,IQ3,MODEL,XVAR,H,MU,PFTL,RHO,U,T,E,SPEC,SUMCI
600  FORMAT(1X,'PSI = ',I4,3X,'IX = ',I4,8X,'SHOCK J COND. = ',I2,14X,
      *'CVD MODEL = ',I2,/,1X,'X = ',D9.4,2X,'H = ',D9.4,2X,'MU = ',
      *E9.4,2X,'P = ',E9.4,3X,'RHO = ',E9.4,/,1X,'U = ',E9.4,2X,'T = ',
      *E9.4,2X,'E = ',E9.4,2X,'CIT = ',E9.4,2X,'SUMCI = ',E9.4 )
      DO 603 I = 3,IMAXP2

```

```

      J = I - 2
      WRITE(*,605) J,VAR(I),J,CM(J),J,VAR(IMAX+I)
603  WRITE(8,605) J,VAR(I),J,CM(J),J,VAR(IMAX+I)
605  FORMAT(6X,'CI(',I2,',') = ',D11.4,4X,'CM(',I2,',') = ',E11.4,4X,
      *       'EVI(',I2,',') = ',D11.4)
C    WRITE(*,620)
      WRITE(8,620)
620  FORMAT(/)
      DO 621 I = 1,IMAX
      IF (EVI(I).EQ.0.0) TVI(I) = 0.0
      IF (I.EQ.IMAX) THEN
        WRITE(*,623) I,TVI(I),TE1,TE2
        WRITE(8,623) I,TVI(I),TE1,TE2
        GO TO 621
      ENDIF
      WRITE(*,622) I,TVI(I)
      WRITE(8,622) I,TVI(I)
621  CONTINUE
622  FORMAT(5X,'TVI(',I2,',') = ',E11.4)
623  FORMAT(5X,'TVI(',I2,',') = ',E11.4,4X,'TE1      = ',E11.4,4X,
      *       'TE2      = ',E11.4)
C    WRITE(*,620)
      WRITE(8,620)
625  IF (NEG.EQ.-1) WRITE(*,630)
630  FORMAT(1X,' A NEGATIVE CONCENTRATION , CI IN MAIN : STOP 670'//)
      IF (NEG.EQ.-1) STOP
C                                                    STOP 670
C WHEN IPSI = (NSR+1), DELX AND STAGNATION STREAMLINES WILL BE COMPUTED
C                               IUNEG = 1  IF U**2 NEG. IN BASIC
      IF (IUNEG.EQ.1) GO TO 735
      IF ((IPSI.LT.(NSR+1)).AND.(VAR(1).LT.XPST(1))) GO TO 505
      IF ((ISTAG.EQ.1).OR.(ISTAG.EQ.2)) GO TO 690
C          ISTAG = 0 UNTIL AFTER EXTRAPOLATION FOR PSI = 0.0 STREAMLINE
C          ISTAG = 1 FOR PSI = 0.0 STREAMLINE
C          ISTAG = 2 FOR DELX STREAMLINE AND THEREAFTER
635  IG = IG+1
      IGC = 0
      IGE = IMAX + 2
      PSIG(IG) = PSIS
      DO 640 IG1 = 3,IMAXP2
      IGC = IGC + 1
      CIG(IGC,IG) = VAR(IG1)
      IGE = IGE + 1
      EIG(IGC,IG) = VAR(IGE)
640  CONTINUE
      EG(IG) = E
      TG(IG) = T
      HG(IG) = H
      IF (IPSI.LT.(NSR+1)) WRITE(8,725)
      IF (IPSI.LT.(NSR+1)) WRITE(*,725)
      IF (IPSI.LT.(NSR+1)) GO TO 730
C                                     IPSI = (NSR+1) IS THE STAGNATION STREAMLINE ;
C                                     AT DELX ,EXTRAPOLATE FOR PSI = 0.0 VALUES
      PSI = 0.0
      MG = 1
      CALL FTLUP (PSI,E,MG,IG,PSIG,EG)
      CALL FTLUP (PSI,T,MG,IG,PSIG,TG)
      CALL FTLUP (PSI,H,MG,IG,PSIG,HG)
      DO 650 IG2 = 1,IMAX
      DO 645 IG1 = 1,NSR1

```

```

      CIT(IG1) = CIG(IG2,IG1)
645 EIT(IG1) = EIG(IG2,IG1)
      CALL FTLUP (PSI,CI(IG2),MG,IG,PSIG,CIT)
      IF (CI(IG2).LT.0.0) CI(IG2) = 1.0E-08
      CALL FTLUP (PSI,EVI(IG2),MG,IG,PSIG,EIT)
650 VAR(2+IMAX+IG2) = EVI(IG2)
C      INITIALIZE FOR PSI = 0.0 STREAMLINE
      CJ = XI
      SPEC = 0.0
      II = 0
      KEYINT = 0
      IX = 0
      SUM = 0.0
      DO 660 I = 1,IMAX
      SUM = SUM + CI(I)/MUI(I)
      IF (EVI(I).EQ.0.0) TVI(I) = 0.0
      IF (EVI(I).EQ.0.0) GO TO 660
      XL = ((DGENI(I)*R*THETAI(I))/(MUI(I)*EVI(I)) + 1.0)
      ALN = DLOG(XL)
      TVI(I) = THETAI(I)/ALN
660 CONTINUE
      MU = 1.0/SUM
      U = DSQRT(2.0*(HSTAG - H))
      VAR(1) = DELX
C      COMPUTE P AND DPDX AT IPSI = (NSR+1) ( AT THE BODY )
      IF (IANS.EQ.0) WRITE(8,665)
      IF ((IPSI.EQ.IS1).OR.(IPSI.EQ.IS2).OR.(IPSI.EQ.IS3)) WRITE(8,665)
      IF ((IPSI.EQ.IS4).OR.(IPSI.EQ.IS5).OR.(IPSI.EQ.IS6)) WRITE(8,665)
      IF (IANS.EQ.0) WRITE(*,665)
      IF ((IPSI.EQ.IS1).OR.(IPSI.EQ.IS2).OR.(IPSI.EQ.IS3)) WRITE(*,665)
      IF ((IPSI.EQ.IS4).OR.(IPSI.EQ.IS5).OR.(IPSI.EQ.IS6)) WRITE(*,665)
665 FORMAT(1X,'-- PSI = 0.000  STREAMLINE --'/)
      P(NSR+1) = PS + US/(RCM(NSR+1)*RSM(NSR+1))*(-PSIS)
      RHO = P(NSR+1)/(H - E)
      NSR2 = NSR + 2
      DO 670 I = NSR2,IZTERM
      READ(10,*) IXT,TS,ES,PS,RHOS,US,PSIS,HS
670 P(I) = PS + US/(RCM(I)*RSM(I))*(-PSIS)
      CLOSE(10)
      CLOSE(11)
      DPDX(NSR+1) = (-3.*P(NSR1)+4.*P(NSR2)-P(NSR3))/(2.*DELX)
C      EVALUATE DPDX FROM (NSR+2) TO IZTERM
      DO 680 I = NSR2,IZTERM
      IF (I.LT.IZTERM) THEN
        DPDX(I) = (- P(I-1) + P(I+1))/(2.*DELX)
      ELSE
        DPDX(I) = (P(I-2) - 4.*P(I-1) + 3.*P(I))/(2.*DELX)
      ENDIF
680 CONTINUE
      OPEN (UNIT=10,FILE='STAG.DAT',STATUS='OLD')
      OPEN (UNIT=11,FILE='EVISC.DAT',STATUS='OLD')
      DO 685 I = 1,NSR
      READ(10,*) IX,TS,ES,PS,RHOS,US,PSIS,HS
685 READ(11,*) (EVIS(J),J = 1,IMAX)
      ISTAG = 1
      PSIS = 0.0
      GO TO 500
C      AT EACH X WHERE PHYSICAL SPACE CALCULATIONS ARE DESIRED SAVE
C      1./(RHO*U) ON EACH STREAMLINE
690 IF (XPST(LPS2).EQ.XPST(1)) ICNT = IPSI

```

```

JJJ = IPSI - ICNT + 1
IF (VAR(1).LT.XPST(LPS2)) GO TO 720
IF (VAR(1).GT.XPST(LPS2)) GO TO 705
C
VAR(1) = XPST(LPS2)
RHOURT = 1.0/(RHO*U)
TEMP = TVI(1)
TEMPO = T
RCON1 = VAR(3)
RCON2 = VAR(4)
RCON3 = VAR(5)
RCON4 = VAR(6)
RCON5 = VAR(7)
RCON6 = VAR(8)
RCON7 = VAR(9)
IF (IMAX.EQ.10) THEN
  RCON8 = VAR(10)
  RCON9 = VAR(11)
  RCON10 = VAR(12)
ENDIF
RRHO = RHO
WRITE(14,*) IPSI,RHOURT,VAR(1),PSIS
WRITE(16,*) LPS2,JJJ
IF (IMAX.EQ.7) WRITE(16,*) RCON1,RCON2,RCON3,RCON4,RCON5,
* RCON6,RCON7,RRHO,TEMP,TEMPO
IF (IMAX.EQ.10) WRITE(16,*) RCON1,RCON2,RCON3,RCON4,RCON5,RCON6
* RCON7,RCON8,RCON9,RCON10,RRHO,TEMP,TEMPO
GO TO 715
705 YPSIS = PREPSI + (XPST(LPS2) - PREX)*((PSIS - PREPSI)/
* (VAR(1) - PREX))
RHOURT = PRERU + (XPST(LPS2) - PREX)*((RHO*U - PRERU)/
* (VAR(1) - PREX))
TEMP = PRETEP + (XPST(LPS2) - PREX)*((TVI(1) - PRETEP)/
* (VAR(1) - PREX))
TEMPO = PRET + (XPST(LPS2) - PREX)*((T - PRET)/
* (VAR(1) - PREX))
RCON1 = PREC1 + (XPST(LPS2) - PREX)*((VAR(3) - PREC1)/
* (VAR(1) - PREX))
RCON2 = PREC2 + (XPST(LPS2) - PREX)*((VAR(4) - PREC2)/
* (VAR(1) - PREX))
RCON3 = PREC3 + (XPST(LPS2) - PREX)*((VAR(5) - PREC3)/
* (VAR(1) - PREX))
RCON4 = PREC4 + (XPST(LPS2) - PREX)*((VAR(6) - PREC4)/
* (VAR(1) - PREX))
RCON5 = PREC5 + (XPST(LPS2) - PREX)*((VAR(7) - PREC5)/
* (VAR(1) - PREX))
RCON6 = PREC6 + (XPST(LPS2) - PREX)*((VAR(8) - PREC6)/
* (VAR(1) - PREX))
RCON7 = PREC7 + (XPST(LPS2) - PREX)*((VAR(9) - PREC7)/
* (VAR(1) - PREX))
IF (IMAX.EQ.10) THEN
  RCON8 = PREC8 + (XPST(LPS2) - PREX)*((VAR(10) - PREC8)/
* (VAR(1) - PREX))
  RCON9 = PREC9 + (XPST(LPS2) - PREX)*((VAR(11) - PREC9)/
* (VAR(1) - PREX))
  RCON10 = PREC10 + (XPST(LPS2) - PREX)*((VAR(12) - PREC10)/
* (VAR(1) - PREX))
ENDIF
RRHO = PRERHO + (XPST(LPS2) - PREX)*((RHO - PRERHO)/
* (VAR(1) - PREX))
RHOURT = 1.0/RHOURT

```

```

WRITE(14,*) IPSI,RHOURT,XPST(LPS2),YPSIS
WRITE(16,*) LPS2,JJJ
IF (IMAX.EQ.7) WRITE(16,*) RCON1,RCON2,RCON3,RCON4,RCON5,
* RCON6,RCON7,RRHO,TEMP,TEMPO
IF (IMAX.EQ.10) WRITE(16,*) RCON1,RCON2,RCON3,RCON4,RCON5,RCON6
* RCON7,RCON8,RCON9,RCON10,RRHO,TEMP,TEMPO
715 LPS2 = LPS2 + 1
KSTAG = KSTAG + 1
720 PREX = VAR(1)
PRERU = RHO*U
PREPSI = PSIS
PRETEP = TVI(1)
PRET = T
PREC1 = VAR(3)
PREC2 = VAR(4)
PREC3 = VAR(5)
PREC4 = VAR(6)
PREC5 = VAR(7)
PREC6 = VAR(8)
PREC7 = VAR(9)
IF (IMAX.EQ.10) THEN
PREC8 = VAR(10)
PREC9 = VAR(11)
PREC10 = VAR(12)
ENDIF
PRERHO = RHO
IF (VAR(1).LT.VARI(IZTERM)) GO TO 505
C END OF STREAMLINE
WRITE(*,725)
WRITE(8,725)
725 FORMAT(1X,'XXXXXXXXXXXXXXXXXX'/)
IF ((IPSI.EQ.(NSR+1)).AND.(ISTAG.EQ.1)) GO TO 425
730 GO TO 420
C COMPUTE PHYSICAL SPACE VALUES
735 WRITE(15,*) KSTAG
CLOSE(10)
CLOSE(11)
CLOSE(12)
CLOSE(13)
CLOSE(15)
CLOSE(16)
WRITE(*,740) KSTAG
WRITE(8,740) KSTAG
740 FORMAT(1X,'BEGIN PHYSICAL SPACE CALCULATIONS KSTAG = ',I5/)
OPEN(UNIT=17,FILE='RADIN3.DAT',STATUS='UNKNOWN')
C
LPS3 = 0
DO 840 LPS = 1,NXPST
CLOSE(14)
OPEN (UNIT=14,FILE='RHORT.DAT',STATUS='OLD')
IK = 0
745 LPS3 = LPS3 + 1
IF (LPS3.GT.IZTERM) GO TO 840
IF (DABS(X1(LPS3) - XPST(LPS)).GT.1.0E-08) GO TO 745
DO 780 IPS = 1,KSTAG
READ(14,*) IPSI,RHOURT,VAR(1),PSIS
VARMX = VAR(1) - XPST(LPS)
IF (DABS(VARMX).GT.1.0E-08) GO TO 780
IK = IK + 1
IBOB = IK - 1

```

```

      IF (IK.GT.200) WRITE(*,770) IK
770 FORMAT(1X,'IK.GT.200,CHANGE DIMENSION OF RUT AND PSISTG,IK = ',I3)
      RUT(IK) = RHOVRT
      PSISTG(IK) = PSIS
780 CONTINUE
C                                     FIND SMALLEST DELTA PSI
      SMALL = 200.0
      DO 785 I = 2,IK
      IF ((PSISTG(I) - PSISTG(I-1)).LT.SMALL) SMALL = PSISTG(I)
      * - PSISTG(I-1)
785 CONTINUE
      XDEL = PSISTG(IK)/SMALL
C                                     TRUNCATE
      L = INT(XDEL + 1.0)
C                                     REMAINDERING   M = 0 FOR EVEN, M = 1 FOR ODD
      M = MOD(L,2)
C                                     MAKE L EVEN
      IF (M.NE.0) L = L + 1
C                                     FIND INTEGRAL 1./(RHO*U) DELTA PSI FROM BODY TO SHOCK
C                                     USING SIMPSONS RULE ; L INCREMENTS , L+1 POINTS
      FL = L
      DPSIS = PSISTG(IK)/FL
      PSII = 0.0
      RB = RUT(1) + RUT(IK)
      DO 800 I = 2,L,2
      PSII = PSII + DPSIS
      CALL FTLUP (PSII,RU1,1,IK,PSISTG,RUT)
      IF (I.EQ.L) RB = RB + 4.0*RU1
      IF (I.EQ.L) GO TO 800
      PSII = PSII + DPSIS
      CALL FTLUP (PSII,RU2,1,IK,PSISTG,RUT)
      RB = RB + 4.0*RU1 + 2.0*RU2
800 CONTINUE
      RB = RB*DPSIS/3.0
      ARR = DSQRT(RSM(LPS3)**2 - 2.0*COSTM(LPS3)*RB)
      YI = (RSM(LPS3) - ARR)/COSTM(LPS3)
      WRITE(17,*) IBOB,IK
      WRITE(17,*) YI
      ZE = ZSM(LPS3) + YI*SINTM(LPS3)
      WRITE(*,815) PSISTG(1),X1(LPS3),ARR,YI,ZE
      WRITE(8,815) PSISTG(1),X1(LPS3),ARR,YI,ZE
815 FORMAT(1X,'PSI = ',E12.5,/,1X,'X = ',E13.6,3X,'R = ',E13.6
      * ,3X,'Y = ',E13.6,3X,'Z = ',E13.6/)
      DO 830 I = 2,IK
      DPSI = PSISTG(I) - PSISTG(I-1)
C                                     TRAPEZOIDAL RULE
      TR = (DPSI/2.0)*(RUT(I) + RUT(I-1))
      RB = RB - TR
      ARR = DSQRT(RSM(LPS3)**2 - 2.0*COSTM(LPS3)*RB)
      YI = (RSM(LPS3) - ARR)/COSTM(LPS3)
      WRITE(17,*) YI
      ZE = ZSM(LPS3) + YI*SINTM(LPS3)
      WRITE(8,815) PSISTG(I),X1(LPS3),ARR,YI,ZE
      WRITE(*,815) PSISTG(I),X1(LPS3),ARR,YI,ZE
830 CONTINUE
      WRITE(8,835)
      WRITE(*,835)
835 FORMAT(//)
840 CONTINUE
845 WRITE(8,850)

```

```

      WRITE(*,850)
850  FORMAT(1X,'END THIS CASE , HALLELUJAH '/')
      CLOSE(8)
      CLOSE(14)
      CLOSE(17)
      STOP
      END

```

C

C

C

SUBROUTINE BASIC

C

C

C

C

BASIC CALLED BY CALINTH TO EVALUATE DERIVATIVES H DOT,
C SUB I DOT AND EV SUB I DOT ; DERIVATIVES START IN DER(1)

C

IMPLICIT REAL*8(A-H,O-Z)

COMMON /A1/ M,IX,IMAX,IPSI,MODEL,TE1,TE2

COMMON /A2/ P(200),DPDX(200),VARI(200)

COMMON /A3/ EVIINF(25),THETAI(25),MUI(25),FI(25),DELHI(25),
* CIINF(25),LI(25)

COMMON /A4/ R,MUINF,DELTA,LAMSQ,OMEGSQ,OMEGA,MU,T,U,TINF

COMMON /A5/ EPSIIL(20,25),GIL(20,25),CID(200,4)

COMMON /A6/ VAR(52),CUVAR(52),DER(51)

COMMON /A7/ MJ(50),EJ(50),AJ(50),BJ(50),DIRECT(50)

COMMON /A8/ TVI(25),NI(25),DGENI(25),BI(25)

COMMON /A10/ NUIJ(26,50),AIJ(25,50),NUPIJ(25,50)

COMMON /A11/ SIGIK(25,25),ALPIK(25,25),BETAIK(25,25)

COMMON /A13/ DELX,ZSTERM,IZTERM,NSR,MW,SP,TS

COMMON /A14/ EINF,PINF,RHOINF,VINF,E,JMAX,KEYINT,RHO,HSTAG

COMMON /A15/ PFTL,KITR1,NIP(25),UP(25)

COMMON /A16/ IUNEG,WE(25),WEXE(25),WEYE(25),WEZE(25)

COMMON /A18/ ITNEG,IEXP

DIMENSION PHI(50),SJ(50),DCIJD(50),EVP(100)

DIMENSION CI(25),EVI(25),DCID(25),DEVID(25),EVIBAR(25)

EQUIVALENCE (CUVAR(2),H),(CUVAR(3),CI(1))

EQUIVALENCE (DER(1),DHDX),(DER(2),DCID(1)).

REAL*8 KPJ,KEQ,KJ(50),LAMSQ,MU,MUI,MUIOT,MUINF,ND

C

C

C

C

DER(1) = DHDX

DER(2) = DCID(1)

MUST BE DHDX AS H MAY BE + OR -

C

C

C

C

C

C

DER(1+IMAX) = DCID(IMAX)

DER(1+IMAX+1) = DEVID(1)

DER(1+IMAX+IMAX) = DEVID(IMAX)

C

C

C

C

C

C

CUVAR(1) = X

CUVAR(2) = H,ENTHALPY - IT MUST BE H WHICH MAY BE + OR -

CUVAR(3) = CI(1)

C

C

C

C

C

C

CUVAR(2+IMAX) = CI(IMAX)

CUVAR(2+IMAX+1) = EVI(1)

CUVAR(2+IMAX+IMAX) = EVI(IMAX)

C

C

C

C

C

C

KEYINT = 1 AT END OF 1ST INTERVAL

IF (ITNEG.EQ.1.OR.IEXP.EQ.1) RETURN

IF (M.NE.0) GO TO 10

DO 5 I = 1,IMAX

J1 = I + IMAX + 2

5 CUVAR(J1) = VAR(J1)

C

INTERPOLATE FOR P ACROSS INTEGRATION INTERVAL

```

10 DO 15 I = 1,IMAX
15 EVI(I) = CUVAR(2+IMAX+I)
   MFTL = 1
   NFTL = IZTERM - IPSI + 1
   CALL FTLUP (CUVAR(1),PFTL,MFTL,NFTL,VARI(IPSI),P(IPSI))
C       INTERPOLATE FOR DPDX
   CALL FTLUP (CUVAR(1),DPFTL,MFTL,NFTL,VARI(IPSI),DPDX(IPSI))
   IF (KEYINT.EQ.0) GO TO 95
   SUM = 0.0
   DO 45 I = 1,IMAX
C       USE SHOCK VALUES FOR INITIAL COMPUTATION ON EACH STREAMLINE
   SUM = SUM + CI(I)/MUI(I)
   IF (EVI(I).EQ.0.0) TVI(I) = 0.0
   IF (EVI(I).EQ.0.0) GO TO 45
   XL = ((DGENI(I)*R*THETAI(I))/(MUI(I)*EVI(I)) + 1.0)
   ALN = DLOG(XL)
C
C       COMPUTE TVI
   TVI(I) = THETAI(I)/ALN
45 CONTINUE
   MU = 1.0/SUM
   U2 = 2.0*(HSTAG - H)
   IF (U2.LT.0.0) WRITE(*,50) H
50 FORMAT(1X,'U**2 IS NEG IN BASIC , H = ',E11.4,2X,
* 'END STREAMLINE INTEGRATION')
   IF (U2.LT.0.0) IUNEG = 1
   IF (U2.LT.0.0) RETURN
   U = DSQRT(U2)
C
C       ITERATE FOR E
   KCODE = 0
55 KITR1 = 0
   ICODE = 0
   MAXI = 50
   TOL1 = .001
   TOL2 = .00001
C
   CALL FOFE (MAXI,TOL1,TOL2,ICODE,E)
C
   IF (ICODE.EQ.0) GO TO 75
   IF (ICODE.EQ.1) WRITE(*,60)
60 FORMAT(1X,'* MAXIMUM ITERATION EXCEEDED IN SUB BASIC *: STOP 66'/)
   IF (ICODE.EQ.2) WRITE(*,65)
65 FORMAT(1X,'*** DERIVATIVE = 0 IN SUB BASIC **** : STOP 66'/)
   WRITE(*,70) ICODE
70 FORMAT(1X,'ICODE = ',I2,' IN SUB BASIC '/)
   IF (ICODE.NE.1) STOP
   IF (KCODE.EQ.3) STOP
C       WHEN ICODE = 1, TRY A NEW STARTING E. DO THIS 2 TIMES ;   STOP 66
   KCODE = KCODE + 1
   GO TO 55
75 IF (IEXP.NE.1) GO TO 85
   KCODE = KCODE + 1
   IEXP = 0
   IF (KCODE.EQ.1) GO TO 55
   IF (KCODE.GT.2) RETURN
   E = H - H*1.E-05
   GO TO 55
85 RHO = PFTL/(H - E)
   T = (PFTL*MU)/(RHO*R)
   IF (T.GT.0.0) GO TO 95
   WRITE(*,90) T,RHO,PFTL,H,E,MU

```



```

90 FORMAT(1X,'T NEGATIVE = ',E12.4,1X,'RHO = ',E11.4,1X,'PFTL = ',
*   E11.4,/,1X,'H = ',E11.4,1X,'E = ',E11.4,1X,'MU = ',E11.4,2X,
*   'IN SUB BASIC'/)
E = H - H*1.0E-06
ITNEG = 1
RETURN
95 KEYINT = 1
PI = 4.0*ATAN(1.0)
BC = 1.38054E-16
AVGN = 6.02252E+23

```

```

C                                     COMPUTE DCIDX AND DEVIDX FOR EACH SPECIE I
DO 210 I = 1,IMAX
DCIDX(I) = 0.0
CVSUM1 = 0.0
CVSUM2 = 0.0

```

```

C                                     CORRECTED RELAXATION TIME FOR PARK MODEL
IF (MODEL.EQ.4.AND.I.EQ.1) THEN
ND = RHO*AVGN*CI(I)/MUI(I)
SIGMAV = 10.0E-17*(50000./T)**2
C = DSQRT(8.*AVGN*BC*T/(PI*MUI(I)))
TAUC = 1.0/(C*SIGMAV*ND)
ENDIF

```

```

C                                     FOR EACH REACTION J
DO 190 J = 1,JMAX
ICT = MOD(J,2)

```

```

C                                     II IS THE SELECTED SPECIE FOR REACTION J
KNT = 0

```

```

IF (ICT.EQ.1) THEN
IF (DIRECT(J).EQ.1.0) PHIC = 1.0
IF (DIRECT(J).EQ.2.0) PHIC = 2.0
II = MJ(J)
100 IF (M.EQ.0) GO TO 115
IF (FI(II).NE.0.0) GO TO 120
115 IF (PHIC.EQ.1.0) PHI(J) = 1.0
IF (PHIC.EQ.2.0) PHI(J+1) = 1.0
GO TO 125
120 IF (MODEL.EQ.3.AND.UP(II).NE.0.0) GO TO 122
TEM = DEXP(THETAI(II)/TVI(II))
TEM1 = DEXP(THETAI(II)/T)
TEM2 = DEXP(-NI(II)*(THETAI(II)/TVI(II) - THETAI(II)/T))
TEM3 = DEXP(THETAI(II)/TVI(II) - THETAI(II)/T)
IF (TEM3.EQ.1.0.AND.PHIC.EQ.1.0) PHI(J) = 1.0
IF (TEM3.EQ.1.0.AND.PHIC.EQ.2.0) PHI(J+1) = 1.0
IF (TEM3.EQ.1.0) GO TO 125

```

```

C                                     PHI = COUPLING COEFF.; FOR CVD,CVDV AND PARK MODEL
IF (PHIC.EQ.1.0) PHI(J) = ((1.0-TEM2)/(TEM3-1.0)*(TEM - 1.0)
*   /(TEM1 - 1.0))/NI(II)
IF (PHIC.EQ.2.0) PHI(J+1) = ((1.0-TEM2)/(TEM3-1.0)*(TEM - 1.0)
*   /(TEM1 - 1.0))/NI(II)
GO TO 125
122 USUM = 0.0
TSUM = 0.0
TFSUM = 0.0
TVTSUM = 0.0
TF = 1.0/(1./TVI(II) - 1./T - 1./UP(II))
NIC = NIP(II) + 1
DO 123 N = 1,NIC
XN = N
EEVP = XN - 0.5
EVP(N) = EEVP*(WE(II) + EEVP*(-WEXE(II) + EEVP*(WEYE(II) +

```

```

*      WEZE(II)*EEVP)))
EVP(N) - EVP(N) - EVP(1)
USUM - USUM + DEXP(EVP(N)/UP(II))
TSUM - TSUM + DEXP(-EVP(N)/T)
TFSUM - TFSUM + DEXP(-EVP(N)/TF)
123 TVTSUM - TVTSUM + DEXP(-EVP(N)/TVI(II))
C      COUPLING COEFF. FOR CVDV-PREFERENTIAL MODEL
      IF (PHIC.EQ.1.0) PHI(J) - (TSUM*TFSUM)/(USUM*TVTSUM)
      IF (PHIC.EQ.2.0) PHI(J+1) - (TSUM*TFSUM)/(USUM*TVTSUM)
125 KNT - KNT + 1
      IF (KNT.EQ.1.AND.PHIC.EQ.1.0) THEN
        PHIC - 2.0
        II - MJ(J+1)
        GO TO 100
      ENDIF
      IF (KNT.EQ.1.AND.PHIC.EQ.2.0) THEN
        PHIC - 1.0
        II - MJ(J+1)
        GO TO 100
      ENDIF
C      START OF LOOP TO DETERMINE EQUILIBRIUM CONSTANT
      BETAJ - 0.0
      FJORT - 0.0
      DO 130 K = 1,IMAX
        SUMG - 0.0
        LII - LI(K)
        DO 128 L = 1,LII
128      SUMG - SUMG + GIL(L,K)*DEXP(-EPSIIL(L,K)/T)
C      MUIOT - (CHEMICAL POTENTIAL)/T
      MUIOT - - (BI(K) + (5. + 2.*FI(K))/2.*(DLOG(TINF) +
*      DLOG(T/TINF)) + FI(K)*DLOG(1./(1. - DEXP(-THETAI(K)/T)))
*      + DLOG(SUMG/GIL(1,K))) + DELHI(K)/(R*T)
      BETAJ - BETAJ + (NUPIJ(K,J) - NUIJ(K,J))
130 FJORT - FJORT + (NUPIJ(K,J) - NUIJ(K,J))*MUIOT
C      KPJ - EQUIL. CONST. (IN TERMS OF PARTIAL PRESSURES)
      KPJ - DEXP(-FJORT)
C      KEQ - EQUILIBRIUM CONSTANT
      KEQ - KPJ*(9.8688225E-07*R*T)**(-BETAJ)
C      KJ - RATE CONSTANT (FORW: J = 1,3,. BACK: J = 2,4,.)
      IF (DIRECT(J).EQ.1.0) THEN
        KJ(J) - AJ(J)*T**BJ(J)*DEXP(-EJ(J)/T)
        IF (J.EQ.19) KJ(J) - AJ(J)*TVI(1)**BJ(J)*DEXP(-EJ(J)/TVI(1))
        KJ(J+1) - KJ(J)/KEQ
      ELSE
        KJ(J+1) - AJ(J)*T**BJ(J)*DEXP(-EJ(J)/T)
        IF (J.EQ.19) KJ(J+1) - AJ(J)*TVI(1)**BJ(J)*DEXP(-EJ(J)/TVI(1))
        KJ(J) - KJ(J+1)*KEQ
      ENDIF
    ENDIF
    IF (NUIJ(IMAX+1,J).EQ.0) SJ(J) - 1.0
    IF (NUIJ(IMAX+1,J).EQ.0) GO TO 140
    SUM - 0.0
    DO 135 ISUM = 1,IMAX
135 SUM - SUM + AIJ(ISUM,J)*CI(ISUM)/MUI(ISUM)
    SJ(J) - RHO*SUM
140 PROD - 1.0
    DO 180 IPROD = 1,IMAX
    IF (NUIJ(IPROD,J).EQ.0) GO TO 180
    TEM - RHO*CI(IPROD)/MUI(IPROD)
    IF (TEM.GE.0.0) GO TO 165

```

```

C LET PROBLEM COMPUTE PROD. WHEN CI NEG ; DECIDE IN MAIN ABOUT ACCEPTING NEG CI
  LABW = MOD(NUIJ(IPROD,J),2)
C
  LABW = 0 FOR EVEN NUIJ, ABW = 1 FOR ODD NUIJ
  IF (LABW.EQ.0) PROD = PROD*(-TEM)**NUIJ(IPROD,J)
  IF (LABW.EQ.1) PROD = -1.0*PROD*(-TEM)**NUIJ(IPROD,J)
  GO TO 180
165 PROD = PROD*TEM**NUIJ(IPROD,J)
180 CONTINUE
  IF (ICT.EQ.1) PRODF = PROD
  BETAIJ = NUPIJ(I,J) - NUIJ(I,J)
  DCIJDX(J) = PHI(J)*KJ(J)*SJ(J)*(MUI(I)/(RHO*U))*PROD*BETAIJ
C
  DCIDX = RATE OF PRODUCTION OF THE CONCENTRATION OF SPECIE I
  DCIDX(I) = DCIDX(I) + DCIJDX(J)
  IF (ICT.EQ.0.AND.MODEL.GT.1) THEN
C
  QIJ = NET RATE OF PRODUCTION
  QIJ = DCIJDX(J-1) + DCIJDX(J)
  DENOM = KEQ*PHI(J-1)*SJ(J-1)*PRODF
  IF (DENOM.EQ.0.0) GO TO 190
C
  CHI = DEGREE OF NON-EQUILIBRIUM
  CHI = 1.0 - (PHI(J)*SJ(J)*PROD/DENOM)
  DENOM1 = CI(I)*CHI
  IF (DENOM1.EQ.0.0) GO TO 190
  CVSUM1 = CVSUM1 + QIJ/DENOM1
  CVSUM2 = CVSUM2 + QIJ*(1.0 - CHI)/DENOM1
  ENDIF
190 CONTINUE
C
  TEM = DEXP(THETAI(I)/T)
  TEM1 = (FI(I)*DGENI(I)*R*THETAI(I))/MUI(I)
  EVIBAR(I) = TEM1/(TEM - 1.0)
  IF (M.NE.0) GO TO 195
  VAR(I+IMAX+2) = EVIBAR(I)
  DEVIDX(I) = 0.0
  GO TO 210
195 TAUSUM = 0.0
  IF (FI(I).EQ.0.0) GO TO 210
C
  CVD MODEL
  DO 200 K = 1,IMAX
  TEM3 = DEXP(-THETAI(I)/T)
  TEM4 = DEXP(SIGIK(K,I)*T**(-1./3.))
  TEM5 = (FI(I)*ALPIK(K,I))/PFTL
  TAUIK = TEM5*(T**BETAIK(K,I)*TEM4)/(1.0 - TEM3)
C
  MILLIKAN & WHITE RELAXATION DATA
  IF (MW.EQ.1) THEN
  IF (I.EQ.1) TAUIK = 1.9E-05*DEXP(216.45*T**(-1./3.))/PFTL
  IF (I.EQ.8) TAUIK = 3.1845E-05*DEXP(198.66*T**(-1./3.))/PFTL
  ENDIF
  TEM6 = CI(K)*(EVIBAR(I) - EVI(I))
C
  PARK MODEL
  IF (MODEL.EQ.4.AND.I.EQ.1) THEN
  TAUPL = TAUIK + TAUC
  PF = DABS((T - TVI(I))/(TS - TINF))** (SP - 1.)
  TAUSUM = TAUSUM + (TEM6/(TAUPL*U))*PF
  GO TO 200
  ENDIF
  TAUSUM = TAUSUM + TEM6/(TAUIK*U)
200 CONTINUE
  IF (MODEL.EQ.1) GO TO 207
  IF (MODEL.EQ.3) GO TO 202
C
  ADDITIONAL TERMS FOR CVDV & PARK MODEL

```

```

      IF (I.EQ.5.OR.I.EQ.6.OR.I.EQ.8) GO TO 207
      TEM7 = DEXP(THETA(I)/TVI(I) - THETA(I)/T)
      TEM8 = DEXP(NI(I)*THETA(I)/TVI(I))*DEXP(-NI(I)*THETA(I)/T)
      CVDVT = ((THETA(I)/(TEM7 - 1.) - NI(I)*THETA(I)/(TEM8 - 1.))
*      *R/MUI(I) - EVI(I))*CVSUM1 - ((0.5*(NI(I) - 1.)*THETA(I)
*      *R/MUI(I)) - EVI(I))*CVSUM2
      TAUSUM = TAUSUM + CVDVT
      GO TO 207

```

C ADDITIONAL TERMS FOR CVDV-PREFERENTIAL MODEL

```

202 IF (I.EQ.5.OR.I.EQ.6.OR.I.EQ.8) GO TO 207
      USUM1 = 0.0
      USUM2 = 0.0
      TFSUM1 = 0.0
      TFSUM2 = 0.0
      NIC = NIP(I) + 1
      TF = 1.0/(1./TVI(I) - 1./T - 1./UP(I))
      DO 204 N = 1,NIC
      XN = N
      EEVP = XN - 0.5
      EVP(N) = EEVP*(WE(I) + EEVP*(-WE(XE(I) + EEVP*(WEYE(I) +
*      WEZE(I)*EEVP)))
      EVP(N) = EVP(N) - EVP(1)
      USUM1 = USUM1 + DEXP(EVP(N)/UP(I))
      USUM2 = USUM2 + EVP(N)*DEXP(EVP(N)/UP(I))
      TFSUM1 = TFSUM1 + DEXP(-EVP(N)/TF)
204 TFSUM2 = TFSUM2 + EVP(N)*DEXP(-EVP(N)/TF)
      GBAR = (USUM2/USUM1)*(R/MUI(I))
      EBAR = (TFSUM2/TFSUM1)*(R/MUI(I))
      CVDVPT = (EBAR - EVI(I))*CVSUM1 - (GBAR - EVI(I))*CVSUM2
      TAUSUM = TAUSUM + CVDVPT

```

C DEVIDX = EQUILIBRIUM VIBRATIONAL ENERGY

```

207 DEVIDX(I) = TAUSUM
      DER(1+IMAX+I) = DEVIDX(I)
210 CONTINUE

```

C COMPUTE DHDX

```

      DHDX = DPFTL/RHO
      RETURN
      END

```

C

C

C

SUBROUTINE FOFE (MAXI,TOL1,TOL2,ICODE,E1)

C

C

C

CALL BY SUB BASIC TO EVALUATE E BY NEWTON ITERATION TECHNIQUE

C

```

      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /A1/ M,IX,IMAX,IPSI,MODEL,TE1,TE2
      COMMON /A2/ P(200),DPDX(200),VARI(200)
      COMMON /A3/ EVIINF(25),THETA(25),MUI(25),FI(25),DELHI(25),
*      CIINF(25),LI(25)
      COMMON /A4/ R,MUINF,DELTA,LAMSQ,OMEGSQ,OMEGA,MU,T,U,TINF
      COMMON /A5/ EPSIIL(20,25),GIL(20,25),CID(200,4)
      COMMON /A6/ VAR(52),CUVAR(52),DER(51)
      COMMON /A7/ MJ(50),EJ(50),AJ(50),BJ(50),DIRECT(50)
      COMMON /A8/ TVI(25),NI(25),DGENI(25),BI(25)
      COMMON /A10/ NUIJ(26,50),AIJ(25,50),NUIPIJ(25,50)
      COMMON /A15/ PFTL,KITR1,NIP(25),UP(25)
      COMMON /A18/ ITNEG,IEXP
      DIMENSION CI(25)
      EQUIVALENCE (CUVAR(3),CI(1)),(CUVAR(2),H)

```

REAL*8 MUI,LAMSQ,MUINF,MU

C

```
IF (IEXP.EQ.1.OR.ITNEG.EQ.1) RETURN
KITR1 = KITR1 + 1
ITER = 0
PORHO = H - E1
T = PORHO*MU/R
1 ITER = ITER + 1
E = 0.0
DE = 0.0
DO 35 I = 1,IMAX
IF (TVI(I).EQ.0.0) THEN
  TEM1 = (1.5*R*T)/MUI(I) + (FI(I)*R*T)/MUI(I)
  DTEM1 = 1.5*R/MUI(I) + FI(I)*R/MUI(I)
  GO TO 15
ENDIF
TEM = DEXP(THETAI(I)/TVI(I))
TEM1 = (1.5*R*T)/MUI(I) + (FI(I)*R*T)/MUI(I) +
* (FI(I)*R*THETAI(I))/(MUI(I)*(TEM - 1.0))
DTEM1 = 1.5*R/MUI(I) + FI(I)*R/MUI(I) + (R*FI(I)*THETAI(I)**2)
* /(MUI(I)*TVI(I)**2)*(TEM/(TEM - 1.0)**2)
15 SUMG = 0.0
SUMGE = 0.0
DSUMGE = 0.0
LII = LI(I)
```

C

```
DO 25 L = 1,LII
TEM3 = - EPSIIL(L,I)/T
IF (TEM3.LT.741.67) GO TO 22
IEXP = 1
RETURN
22 TEM2 = DEXP(TEM3)
SUMG = SUMG + GIL(L,I)*TEM2
SUMGE = SUMGE + GIL(L,I)*EPSIIL(L,I)*TEM2
IF (SUMGE.LT.1.0D+34) GO TO 25
IEXP = 1
RETURN
25 DSUMGE = DSUMGE + TEM2*GIL(L,I)*EPSIIL(L,I)**2
DEVI = R/MUI(I)*(((SUMG*DSUMGE/T**2) - (SUMGE/T)**2)
* /SUMG**2)
DEI = DTEM1 + DEVI
EI = TEM1 + R/MUI(I)*(SUMGE/SUMG) + DELHI(I)/MUI(I)
DE = DE + DEI*CI(I)
E = E + EI*CI(I)
35 CONTINUE
```

C

```
IF (E.GT.H) ITNEG = 1
DFT = - DE
FT = E1 - E
IF (DFT.EQ.0.0) THEN
  ICODE = 2
  RETURN
ENDIF
TS1 = T - (FT/DFT)
DELT = DABS(TS1 - T)
T = TS1
E1 = H - R*T/MU
IF (ITER.GE.MAXI) THEN
  ICODE = 1
  RETURN
```

```

ENDIF
IF (DELT.GT.TOL1) THEN
  ITER1 = ITER
  GO TO 1
ENDIF
ITER2 = ITER1 + 10
IF (DELT.LE.TOL2.OR.ITER.GE.ITER2) RETURN
GO TO 1
END

C
C-----
C
  SUBROUTINE FOFTS1 (MAXI,TOL1,TOL2,ICODE,TSG)
C
C  CALLED BY MAIN TO EVALUATE TS BY NEWTON ITERATION TECHNIQUE
C
  IMPLICIT REAL*8(A-H,O-Z)
  COMMON /A1/ M,IX,IMAX,IPSI,MODEL,TE1,TE2
  COMMON /A2/ P(200),DPDX(200),VARI(200)
  COMMON /A3/ EVIINF(25),THETAI(25),MUI(25),FI(25),DELHI(25)
  *      ,CIINF(25),LI(25)
  COMMON /A4/ R,MUINF,DELTA,LAMSQ,OMEGSQ,OMEGA,MU,T,U,TINF
  COMMON /A5/ EPSIIL(20,25),GIL(20,25),CID(200,4)
  COMMON /A6/ VAR(52),CUVAR(52),DER(51)
  REAL*8 MUI,LAMSQ,MUINF,MU

C
  ITER = 0
1  ITER = ITER + 1
  ES = 0.0
  DES = 0.0
  DO 25 I = 1,IMAX
    IF (M.EQ.1) EVIS = EVIINF(I)
    IF (M.EQ.1) GO TO 5
    TEM = DEXP(THETAI(I)/TSG)
    EVIS = (R*THETAI(I))/(MUI(I)*(TEM-1.0))*FI(I)
    DEVIS = (R*FI(I)*THETAI(I)**2)/(MUI(I)*TSG**2)*(TEM/(TEM-1.0)**2)
5  SUMG = 0.0
  SUMGE = 0.0
  DSUMGE = 0.0
  LII = LI(I)

C
  DO 10 L = 1,LII
    TEM1 = DEXP(- EPSIIL(L,I)/TSG)
    SUMG = SUMG + TEM1*GIL(L,I)
    SUMGE = SUMGE + TEM1*GIL(L,I)*EPSIIL(L,I)
10  DSUMGE = DSUMGE + TEM1*GIL(L,I)*EPSIIL(L,I)**2
    EEIS = (R/MUI(I))*(SUMGE/SUMG)
    DEEIS = R/MUI(I)*(((SUMG*DSUMGE/TSG**2) - (SUMGE/TSG)**2)
  *      /SUMG**2)
    EIS = (1.5*R*TSG)/MUI(I) + (FI(I)*R*TSG)/MUI(I) + EVIS + EEIS
  *      + DELHI(I)/MUI(I)
    DEIS = 1.5*R/MUI(I) + FI(I)*R/MUI(I) + DEVIS + DEEIS
    ES = ES + EIS*CIINF(I)
    DES = DES + DEIS*CIINF(I)
25  CONTINUE

C
  FAC = DSQRT(OMEGSQ - 2.0*LAMSQ*(DELTA - ES))
  FIN = (2.0*MUINF*(DELTA - ES)*FAC)/(R*(OMEGA + FAC))
  FTS = TSG - FIN
  FAC1 = R*(OMEGA+FAC)*(MUINF*(DELTA-ES)*(2.0/FAC)*LAMSQ*DES -

```

```

*      2.0*FAC*MUINF*DES) - 2.0*MUINF*(DELTA-ES)*R*LAMSQ*DES
FAC2 = R**2*(OMEGA + FAC)**2
DFTS = 1.0 - FAC1/FAC2
IF (DFTS.EQ.0.0) THEN
  ICODE = 2
  RETURN
ENDIF
TSG1 = TSG - (FTS/DFTS)
DELTS = DABS(TSG1 - TSG)
TSG = TSG1
IF (ITER.GE.MAXI) THEN
  ICODE = 1
  RETURN
ENDIF
IF (DELTS.GT.TOL1) THEN
  ITER1 = ITER
  GO TO 1
ENDIF
ITER2 = ITER1 + 10
IF (DELTS.LE.TOL2.OR.ITER.GE.ITER2) RETURN
GO TO 1
END

```

C
C-----
C

SUBROUTINE CHECK

C
C
C
C
C
C

CHECK CALLED BY CALINTH TO MAKE DECISION TO ACCEPT ANSWERS

IF ACCEPTABLE ; SET ELB = 0 AND RETURN

IF NOT ACCEPTABLE ; MODIFY SPEC AND CI ; SET ELB = 1.0 AND RETURN

IMPLICIT REAL*8(A-H,O-Z)

COMMON /A1/ M,IX,IMAX,IPSI,MODEL,TE1,TE2

COMMON /A2/ P(200),DPDX(200),VARI(200)

COMMON /A3/ EVIINF(25),THETAI(25),MUI(25),FI(25),DELHI(25),

* CIINF(25),LI(25).

COMMON /A4/ R,MUINF,DELTA,LAMSQ,OMEGSQ,OMEGA,MU,T,U,TINF

COMMON /A5/ EPSIIL(20,25),GIL(20,25),CID(200,4)

COMMON /A6/ VAR(52),CUVAR(52),DER(51)

COMMON /A13/ DELX,ZSTERM,IZTERM,NSR,MW,SP,TS

COMMON /A17/ ELB,SPEC,CJ,TPREV,HPREV,HCHECK,TCHECK

COMMON /A18/ ITNEG,IEXP

REAL*8 MUI,MU,MUINF,LAMSQ

EQUIVALENCE (CUVAR(2),H)

C

IF (IEXP.NE.1) GO TO 1

GO TO 15

C

REDUCE COMPUTING INTERVAL TO TRY AND AVOID EXP ERROR STOP

1 IF (ITNEG.EQ.0) GO TO 4

C

E.GT.H SOMETIMES LEADS TO NEG T--REDUCE INTERVAL TO TRY TO AVOID INSTABILITY
GO TO 15

4 IMAXP1 = IMAX + 1

DO 10 I = 3,IMAXP1

C

CUVAR(2) = H MAY BE NEGATIVE

IF (CUVAR(I).LT.0.0) WRITE(*,5) I,IX,CUVAR(I)

IF (CUVAR(I).LT.0.0) GO TO 15

5 FORMAT(1X,'NEG CI AT I = ',I3,1X,'IX = ',I3,1X,'CUVAR(I) = ',

* E12.5,1X,'IN SUB CHECK'/)

10 CONTINUE

```

      IF (IX.LT.3) GO TO 30
      IF ((IPSI.EQ.(NSR+1)).AND.(IX.LT.(NSR+3))) GO TO 30
      IF (DABS(TPREV - T)/T.LT.TCHECK) GO TO 25
      WRITE(*,11)
11  FORMAT(1X,'CHECK ABS(TPREV-T)/T.GE.TCHECKT IN SUB CHECK'/)
15  IF (SPEC.GT.1.0E-15) GO TO 20
      WRITE(*,*)'SPEC .LT. 1.0E-15 IN SUB CHECK'          STOP 30'
      STOP
C
C                                     REDUCE INTERVAL
C
20  SPEC = SPEC/4.0
      CJ = SPEC
      ELB = 1.0
      WRITE(*,22) SPEC,IPSI,IX,T,TPREV,H,HPREV,IEXP,ITNEG
22  FORMAT(1X,'REDUCED SPEC = ',E9.4,2X,'IPSI = ',I3,2X,'IX = ',I3,
* 2X,'T = ',E9.4,2X,'TPREV = ',E9.4,/,1X,'H = ',E9.4,3X,
* 'HPREV = ',E9.4,3X,'IEXP = ',I3,3X,'ITNEG = ',I3,' IN CHECK'/)
      ITNEG = 0
      IEXP = 0
      RETURN
25  IF (DABS((HPREV - H)/H).GT.HCHECK) WRITE(*,27)
27  FORMAT(1X,'CHECK ABS((HPREV-H)/H).GT.HCHECKT IN SUB CHECK'/)
      IF (DABS((HPREV - H)/H).GT.HCHECK) GO TO 15
C
C                                     ACCEPTABLE
C
30  TPREV = T
      HPREV = H
      ELB = 0.0
      RETURN
      END
C
C-----
C
      SUBROUTINE SHOCKG (NXPSTM1,ITK)
C
C  SUBROUTINE SHOCKG CALLED BY MAIN AND MUST BE SUPPLIED BY USER.
C  X,ZS,RS,RC,COST,SINT FOR EACH X FROM 0.0 TO X AT ZSTERM IN INCREMENTS
C  OF (DELX/NSR) TO DELX AND INCREMENTS OF DELX THEREAFTER.
C
C      X          -   DISTANCE ALONG SHOCK
C      ZS         -   DISTANCE ALONG SHOCK AXIS OF SYMMETRY
C      RS         -   RADIUS OF SHOCK
C      RC         -   RADIUS OF CURVATURE OF SHOCK
C      COST       -   COS OF ANGLE OF ATTACK
C      SINT       -   SIN OF ANGLE OF ATTACK
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /A9/  EVIS(25),XPST(100)
      COMMON /A12/ SINTM(200),COSTM(200),RSM(200),RCM(200),X1(200)
*           ,ZSM(200)
      COMMON /A13/ DELX,ZSTERM,IZTERM,NSR,MW,SP,TS
      DIMENSION ZDUM(501),RSDUM(501),RCDUM(501),XC(501),CODUM(501),
*           SIDUM(501)
C
C                                     Rs/L = SQRT(2*R*Zs/L - Bs*(Zs/L)**2)   WHERE: L = 1.0 cm
C
      ITK = 0
      EL = 1.0
      R = 230.0
      BS = -4.0
      DO 5 I = 2,501
5  ZDUM(I) = (I-1)*ZSTERM/500
      ZDUM(1) = 0.0

```



```

RSDUM(1) = 0.0
XC(1) = 0.0
CODUM(1) = 0.0
SIDUM(1) = 1.0
RCDUM(1) = 230.0

C
DO 10 I = 2,501
ZSND = ZDUM(I)/EL
RSDUM(I) = DSQRT(2.*R*ZSND - BS*ZSND**2)
C = 1.0/(R - BS*ZSND)**2
C1 = DSQRT(C*RSDUM(I)**2 + 1.0)
XC(I) = RSDUM(I)/2.*C1 + 1./(2.*DSQRT(C))*DLOG(RSDUM(I)*DSQRT(C)
*      + C1)
D1 = (R - BS*ZSND)/RSDUM(I)
D2 = - (R - BS*ZSND)**2/RSDUM(I)**3 - BS/RSDUM(I)
THETA = ATAN(D1)
SIDUM(I) = DSIN(THETA)
CODUM(I) = DCOS(THETA)
RCDUM(I) = (1.0 + D1**2)**1.5/DABS(D2)

C                                     DIMENSIONALIZED QUANTITIES
RCDUM(I) = EL*RCDUM(I)
XC(I) = EL*XC(I)
RSDUM(I) = EL*RSDUM(I)
10 CONTINUE

C
WRITE(*,25)
WRITE(8,25)
25 FORMAT(5X,'X',11X,'ZS',10X,'RS',10X,'RC',9X,'COST',8X,'SINT'/)
NOX = 0
M = 1
NZS = 501

C                                     INTERPOLATE SHOCK VALUES TO X SHOCK COORDINATE
30 NOX = NOX + 1
IF (NOX.LT.(NSR+2)) X = X + DELX/NSR
IF (NOX.EQ.1) X = 0.0
IF (NOX.GT.(NSR+1)) X = X + DELX
CALL FTLUP (X,ZS,M,NZS,XC,ZDUM)
CALL FTLUP (X,RS,M,NZS,XC,RSDUM)
CALL FTLUP (X,RC,M,NZS,XC,RCDUM)
CALL FTLUP (X,COST,M,NZS,XC,CODUM)
CALL FTLUP (X,SINT,M,NZS,XC,SIDUM)
WRITE(8,35) X,ZS,RS,RC,COST,SINT,NOX
WRITE(*,35) X,ZS,RS,RC,COST,SINT,NOX
35 FORMAT(1X,6(E10.4,2X),I4)
SINTM(NOX) = SINT
COSTM(NOX) = COST
RSM(NOX) = RS
RCM(NOX) = RC
X1(NOX) = X
ZSM(NOX) = ZS
DO 36 IT = 2,NXPSTM1
IF (DABS(XPST(IT) - X).GT.1.0E-06) GO TO 36
ITK = ITK + 1
36 CONTINUE
IF (ZS.GE.0.0) GO TO 40
WRITE(*,37) ZS
37 FORMAT(1X,'ZS SHOULD BE GREATER THAN 0.0 : ZS = ',E10.5,
*      /,1X,'STOP 13'/)
STOP

C                                     STOP 13

```

```

40 IF (ZS.LT.ZSTERM) GO TO 30
   IZTERM = NOX
   RETURN
   END

```

C

C-----

C

```

SUBROUTINE CALINTH (N,CIMAX,PHMAX)

```

C

C

C

C

C

C

```

IN THE CALINTH VERSION OF CALINT, THE VARIABLE IN VAR(2) AND
CUVAR(2) MAY BE + OR -. VALUES OF OTHER DEPENDENT VARIABLES
ARE EXPECTED TO BE POSITIVE.          10-69

```

```

IMPLICIT REAL*8(A-H,O-Z)

```

```

COMMON /A1/ M,IX,IMAX,IPSI,MODEL,TE1,TE2

```

```

COMMON /A2/ P(200),DPDX(200),VARI(200)

```

```

COMMON /A3/ EVIINF(25),THETAI(25),MUI(25),FI(25),DELHI(25),

```

```

*      CIINF(25),LI(25)

```

```

COMMON /A4/ R,MUINF,DELTA,LAMSQ,OMEGSQ,OMEGA,MU,T,U,TINF

```

```

COMMON /A5/ EPSIIL(20,25),GIL(20,25),CID(200,4)

```

```

COMMON /A6/ VAR(52),CUVAR(52),DER(51)

```

```

COMMON /A7/ MJ(50),EJ(50),AJ(50),BJ(50),DIRECT(50)

```

```

COMMON /A8/ TVI(25),NI(25),DGENI(25),BI(25)

```

```

COMMON /A10/ NUIJ(26,50),AIJ(25,50),NUPIJ(25,50)

```

```

COMMON /A11/ SIGIK(25,25),ALPIK(25,25),BETAIK(25,25)

```

```

COMMON /A17/ ELB,SPEC,CJ,TPREV,HPREV,HCHECK,TCHECK

```

```

COMMON /A19/ ELE1(51),ELE2(51),NERR

```

```

REAL*8 MUI,MU,MUINF,LAMSQ

```

```

DIMENSION F1(51),F2(51),F3(51),CAPF1(51),CAPF2(51),CAPF3(51)

```

```

*      ,P1(51),PH(51),DELTY(51),Y3(51),Y4(51),F4(51),Y2(51)

```

C

C

C

C

C

```

IF CJ OR N IS EQUAL TO ZERO :          NERR = 1

```

```

IF ELE1 IS LESS THAN OR EQUAL TO ELE2 : NERR = 2

```

TEST INPUT

```

FN = N

```

```

TEST = CJ*FN

```

```

IF (TEST) 998,997,998

```

```

997 NERR = 1

```

```

RETURN

```

```

998 DO 999 I = 1,N

```

```

IF ((ELE1(I) - ELE2(I)).LE.0.0) THEN

```

```

  NERR = 2

```

```

  RETURN

```

```

ENDIF

```

```

999 CONTINUE

```

```

1000 IF (SPEC) 5,1,5

```

C

SECTION FOR INITIALIZATION COMPUTATION OF DERIVATIVES

```

1 SPEC = CJ

```

```

  ICONT = 1

```

```

2 N1 = N + 1

```

```

  DO 3 I = 1,N1

```

```

3 CUVAR(I) = VAR(I)

```

```

  CALL BASIC

```

C

RETURN WITH DERIVATIVES IN DER

```

DO 4 I = 1,N

```

```

4 F1(I) = DER(I)

```

```

  RETURN

```

C

COMPUTE Y2,X2

```

5 CUVAR(1) = VAR(1) + CJ/2.0

```

```

  DO 6 I = 1,N

```

```

        I1 = I + 1
        Y2(I) = VAR(I1) + CJ/2.0*F1(I)
        IF (I.EQ.1) GO TO 6
        IF (Y2(I)) 65,6,6
    6   CUVAR(I1) = Y2(I)
        GO TO 66
    65  SPEC = CJ
        CJ = CJ/2.0
        GO TO 5
C
                                CALL BASIC TO EVALUATE F2
    66  CALL BASIC
C
                                RETURN
        DO 7 I = 1,N
        I1 = I + 1
        F2(I) = DER(I)
C
                                COMPUTE Y3
        Y3(I) = VAR(I1) + CJ/2.0*F2(I)
        IF (I.EQ.1) GO TO 7
        IF (Y3(I)) 65,7,7
    7   CUVAR(I1) = Y3(I)
C
                                CALL BASIC TO EVALUATE F3
        CALL BASIC
C
                                RETURN
        DO 10 I = 1,N
        F3(I) = DER(I)
C
                                COMPUTE P,PH AND CAP F TERMS
        IF (Y3(I) - Y2(I)) 9,8,9
    8   P1(I) = 0.0
        GO TO 91
    9   P1(I) = -((F3(I) - F2(I))/(Y3(I) - Y2(I)))
    91  PH(I) = P1(I)*CJ
        IF (PH(I)) 83,83,103
    83  PH(I) = 0.0
        P1(I) = 0.0
        GO TO 84
    103 Z1 = DABS(Y3(I) - Y2(I))/((DABS(Y3(I)) + DABS(Y2(I)))/2.0)
        IF (Z1 - 0.5E-04) 83,83,84
    84  IF (PH(I) - 0.1) 85,85,95
    85  CAPF1(I) = 1.0 - PH(I)/2.0 + PH(I)**2/6.0 - PH(I)**3/24.0
        CAPF2(I) = 0.5 - PH(I)/6.0 + PH(I)**2/24.0 - PH(I)**3/120.0
        CAPF3(I) = 1./6. - PH(I)/24. + PH(I)**2/120. - PH(I)**3/720.
        GO TO 10
    95  CAPF1(I) = (DEXP(-PH(I)) - 1.0)/(- PH(I))
        CAPF2(I) = (CAPF1(I) - 1.0)/(- PH(I))
        CAPF3(I) = (CAPF2(I) - 0.5)/(- PH(I))
    10  CONTINUE
C
                                IS PH BETWEEN ELE2 AND ELE1
        IF (ICONT - 1) 101,101,102
    102 ICONT = ICONT - 1
        SPEC = CJ
        GO TO 17
    101 DO 11 I = 1,N
        IF (DABS(PH(I)) - ELE1(I)) 11,11,13
    11  CONTINUE
        SPEC = CJ
        GO TO 15
C
                                HALVE INTERVAL AND DOUBLE PH RANGE
    13  DO 96 I = 1,N
        ELE1(I) = ELE1(I)*2.0
        IF (ELE1(I) - PHMAX) 94,94,955

```

```

94  ELE2(I) - ELE2(I)*2.0
    GO TO 96
955 ELE1(I) - ELE1(I)/2.0
96  CONTINUE
    SPEC = CJ
    CJ = CJ/2.0
    ICONT = 3
    GO TO 5
C                                     RETURN TO RECOMPUTE INTERVAL
15  DO 16 I = 1,N
    IF (DABS(PH(I)) - ELE2(I)) 16,17,17
16  CONTINUE
C                                     DOUBLE INTERVAL
    CJ = 2.0*CJ
    IF (CJ - CIMAX) 17,17,165
165  CJ = CIMAX
C                                     COMPUTE Y4,X4
17  DO 18 I = 1,N
    I1 = I + 1
    CUVAR(I1) = VAR(I1) + SPEC*(F3(I)*(2.0*CAPF2(I)) + F1(I)*
*      (CAPF1(I) - 2.0*CAPF2(I)) + F2(I)*PH(I)*CAPF2(I))
    IF (I.EQ.1) GO TO 18
    IF (CUVAR(I1)) 175,18,18
175  CJ = SPEC
    CJ = CJ/2.0
    GO TO 5
18  Y4(I) = CUVAR(I1)
    CUVAR(1) = VAR(1) + SPEC
C                                     CALL BASIC TO EVALUATE F4
    CALL BASIC
C                                     RETURN
    DO 20 I = 1,N
    I1 = I + 1
    F4(I) = DER(I)
C                                     COMPUTE DELTA Y
C
    DELTY(I) = SPEC*(F1(I)*CAPF1(I)+(-3.0*(F1(I)+P1(I)*VAR(I1))+2.0*
*      (F2(I)+P1(I)*Y2(I)) + 2.0*(F3(I)+P1(I)*Y3(I)) - F4(I) - P1(I)*
*      Y4(I))*CAPF2(I) + 4.0*((F1(I)+P1(I)*VAR(I1)) - (F2(I)+P1(I)*
*      Y2(I)) - (F3(I)+P1(I)*Y3(I)) + (F4(I)+P1(I)*Y4(I)))*CAPF3(I))
C
C                                     COMPUTE Y + DELTA Y
20  CUVAR(I1) = VAR(I1) + DELTY(I)
C                                     CALL CHECK FOR DECISION TO ACCEPT OR RECOMPUTE INTERVAL
    CALL CHECK
    IF (ELB) 21,21,23
C                                     UPDATE Y VALUES
21  N1 = N + 1
    DO 22 I = 2,N1
    I1 = I - 1
22  VAR(I) = VAR(I) + DELTY(I1)
    VAR(1) = VAR(1) + SPEC
C                                     RETURN TO COMPUTE DERIVATIVES AT Y + DELTA Y
    GO TO 2
C                                     RETURN TO RECOMPUTE INTERVAL
23  GO TO 5
    END
C
C-----
C

```

```

SUBROUTINE FTLUP (X,Y,M,N,VARI,VARD)
C
C THIS SUBROUTINE IS A MODIFICATION OF LIBRARY INTERPOLATION
C SUBROUTINE FTLUP REVISED 7-7-69
C
C IMPLICIT REAL*8(A-H,O-Z)
C DIMENSION VARI(1),VARD(1),V(3),YY(2),II(100)
C INITIALIZE ALL INTERVAL POINTERS TO -1.0 FOR MONOTONICITY CHECK
DO 4 J = 1,100
4 II(J) = -1
MA = IABS(M)
C ASSIGN INTERVAL POINTER FOR GIVEN VARI TABLE ; THE SAME POINTER
C WILL BE USED ON A GIVEN VARI TABLE EVERY TIME
LOCF = VARI(1)
LI = MOD(LOCF,100) + 1
I = II(LI)
IF (I.GE.0) GO TO 10
IF (N.LT.2) GO TO 10
C MONOTONICITY CHECK
IF (VARI(2) - VARI(1)) 1,1,3
C ERROR IN MONOTONICITY
2 K = LOCF
WRITE(*,102) J,K
102 FORMAT(1X,'TABLE BELOW OUT OF ORDER FOR FTLUP AT POSITION ',
* 15,//,1X,'X TABLE IS STORED IN LOCATION',15//)
DO 103 J = 1,N
103 WRITE(*,*) VARI(J),VARD(J)
STOP
C MONOTONIC DECREASING
1 DO 5 J = 2,N
IF (VARI(J) - VARI(J-1)) 5,2,2
5 CONTINUE
GO TO 10
C MONOTONIC INCREASING
3 DO 6 J = 2,N
IF (VARI(J) - VARI(J-1)) 2,2,6
6 CONTINUE
C INTERPOLATION
10 IF (I.LE.0) I = 1
IF (I.GE.N) I = N - 1
IF (N.LE.1) GO TO 8
IF (MA.NE.0) GO TO 99
C ZERO ORDER
8 Y = VARD(1)
GO TO 800
C LOCATE I INTERVAL (X(I).LE.X.LT.X(I+1))
99 IF ((VARI(I) - X)*(VARI(I+1) - X)) 61,61,40
C IN GIVES DIRECTION FOR SEARCH INTERVALS
40 SIGN1 = 1.0
IN = SIGN(SIGN1,(VARI(I+1) - VARI(I))*(X - VARI(I)))
C IF X OUTSIDE ENDPOINTS, EXTRAPOLATE FROM END INTERVAL
41 IF ((I+IN).LE.0) GO TO 61
IF ((I+IN).GE.N) GO TO 61
I = I + IN
IF ((VARI(I) - X)*(VARI(I+1) - X)) 61,61,41
61 IF (MA.EQ.2) GO TO 200
C FIRST ORDER
Y = (VARD(I)*(VARI(I+1) - X) - VARD(I+1)*(VARI(I) - X))/
* (VARI(I+1) - VARI(I))
GO TO 800

```

```

C                                SECOND ORDER
200  IF (N.EQ.2) GO TO 2
      IF (I.EQ.(N-1)) GO TO 209
      IF (I.EQ.1) GO TO 201

C                                PICK THIRD POINT
      SK = VARI(I+1) - VARI(I)
      IF ((SK*(X - VARI(I-1))).LT.(SK*(VARI(I+2) - X))) GO TO 209
201  L = I
      GO TO 702
209  L = I - 1
702  V(1) = VARI(L) - X
      V(2) = VARI(L+1) - X
      V(3) = VARI(L+2) - X
      YY(1) = (VARD(L)*V(2)-VARD(L+1)*V(1))/(VARI(L+1)-VARI(L))
      YY(2) = (VARD(L+1)*V(3)-VARD(L+2)*V(2))/(VARI(L+2)-VARI(L+1))
      Y = (YY(1)*V(3) - YY(2)*V(1))/(VARI(L+2) - VARI(L))
800  II(LI) = I
      RETURN
      END

C
C-----
C
      SUBROUTINE FOFTS23 (MAXI,TOL1,TOL2,ICODE,TSG,IQ1,IQ2,IQ3)
C
C  CALLED BY MAIN TO EVALUATE TS BY NEWTON ITERATION TECHNIQUE
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /A1/ M,IX,IMAX,IPSI,MODEL,TE1,TE2
      COMMON /A2/ P(200),DPDX(200),VARI(200)
      COMMON /A3/ EVIINF(25),THETAI(25),MUI(25),FI(25),DELHI(25),
*      CIINF(25),LI(25)
      COMMON /A4/ R,MUINF,DELTA,LAMSQ,OMEGSQ,OMEGA,MU,T,U,TINF
      COMMON /A5/ EPSIIL(20,25),GIL(20,25),CID(200,4)
      COMMON /A6/ VAR(52),CUVAR(52),DER(51)
      DIMENSION SUMGM(25),EISM(25),DEISM(25)
      REAL*8 MUI,LAMSQ,MUINF,MU

C
      W = 1.0
      DO 1 I = 1,IMAX
1    CID(IX,I) = CIINF(I)
      ITER = 0
2    ITER = ITER + 1
      IF (ITER.EQ.1) THEN
        ES = DELTA
        PS = DSQRT(OMEGSQ - 2.*LAMSQ*(DELTA - ES))
      ENDIF
      IF (IQ1.EQ.1) THEN
        TVIB = TINF
      ELSE
        TVIB = TSG
      ENDIF
      IF (IQ2.EQ.1) THEN
        TEL = TINF
      ELSE
        TEL = TSG
      ENDIF

C
      ES = 0.0
      DES = 0.0
      DO 25 I = 1,IMAX

```

```

IF (M.EQ.1) EVIS = EVIINF(I)
IF (M.EQ.1) GO TO 5
TEM = DEXP(THETAI(I)/TVIB)
EVIS = (R*THETAI(I))/(MUI(I)*(TEM - 1.0))*FI(I)
DEVIS = (R*FI(I)*THETAI(I)**2)/(MUI(I)*TVIB**2)*(TEM/(TEM-1.0)**2)
5 IF (I.EQ.1) QVN = 1.0/(1.0 - DEXP(-THETAI(I)/TVIB))
IF (I.EQ.2) QVO = 1.0/(1.0 - DEXP(-THETAI(I)/TVIB))
SUMGM(I) = 0.0
SUMGE = 0.0
DSUMGE = 0.0
LII = LI(I)

```

C

```

DO 10 L = 1, LII
TEM1 = DEXP(-EPSIIL(L,I)/TEL)
SUMGM(I) = SUMGM(I) + TEM1*GIL(L,I)
SUMGE = SUMGE + TEM1*GIL(L,I)*EPSIIL(L,I)
10 DSUMGE = DSUMGE + TEM1*GIL(L,I)*EPSIIL(L,I)**2
QROTN = TSG/5.8
QROTO = TSG/4.2
EEIS = (R/MUI(I))*(SUMGE/SUMGM(I))
DEEIS = R/MUI(I)*(((SUMGM(I)*DSUMGE/TEL**2) - (SUMGE/TEL)**2)
* /SUMGM(I)**2)
EISM(I) = (1.5*R*TSG)/MUI(I) + (FI(I)*R*TSG)/MUI(I) + EVIS + EEIS
* + DELHI(I)/MUI(I)
DEISM(I) = 1.5*R/MUI(I) + FI(I)*R/MUI(I) + DEVIS + DEEIS
25 CONTINUE

```

C

BETA FOR OXYGEN

```

IF (IQ3.EQ.2) THEN
TD = 59500.0
F1 = CIINF(2)*MUIINF/MUI(2)
C1A = TD/TSG
C1A = DEXP(C1A/2.0)
C1B = 8.8568195*(1. - CIINF(1))*C1A*(1./(TSG**2.5))
C1 = C1A*2.0*MUIINF*PS*C1B*QVO*QROTO*(SUMGM(2)/(SUMGM(4)**2))/R
BETAO = ((F1-1.) + DSQRT((1.-F1)**2 + 4.*(F1+C1)))/(2.*(F1+C1))
BETAN = 0.0
ENDIF

```

C

BETA FOR NITROGEN AND OXYGEN

```

IF (IQ3.EQ.3) THEN
BETAN = .1
BETAO = 1.0
TDN = 113500.0
TDO = 59500.0
F1A = 8.8568195*(1. - CIINF(1))*DEXP(TDO/TSG)*(1./(TSG**2.5))
F2A = 12.36681*(1. - CIINF(2))*DEXP(TDN/TSG)*(1./(TSG**2.5))
F1 = 2.0*MUIINF*PS*F1A*QVO*QROTO*(SUMGM(2)/(SUMGM(4)**2))/R
F2 = 2.0*MUIINF*PS*F2A*QVN*QROTN*(SUMGM(1)/(SUMGM(3)**2))/R
30 A = ((1. - BETAO**2) + BETAN*(1. - BETAO))/(BETAO**2) - F1
B = ((1. - BETAN**2) + BETAO*(1. - BETAN))/(BETAN**2) - F2
ABO = (BETAN*BETAO - 2.*(1. + BETAN))/(BETAO**3)
ABN = (1. - BETAO)/(BETAO**2)
BBO = (1. - BETAN)/(BETAN**2)
BBN = (BETAN*BETAO - 2.*(1. + BETAO))/(BETAN**3)
DEN = ABO*BBN - BBO*ABN
DBO = (-A*BBN + B*ABN)/DEN
DBN = (-B*ABO + A*BBO)/DEN
BETAN = BETAN + DBN
BETAO = BETAO + DBO
IF ((DABS(DBO).GT..00001).OR.(DABS(DBN).GT..00001)) GO TO 30
ENDIF

```

```

C                                     DISSOCIATING CONCENTRATIONS ACROSS THE SHOCK
CID(IX,2) - (1.0 - BETAO)*CIINF(2)
CID(IX,4) - BETAO*CIINF(2)
CID(IX,1) - (1.0 - BETAN)*CIINF(1)
CID(IX,3) - BETAN*CIINF(1)

C
DO 35 I = 1,IMAX
ES = ES + EISM(I)*CID(IX,I)
35 DES = DES + DEISM(I)*CID(IX,I)
FAC = DSQRT(OMEGSQ - 2.0*LAMSQ*(DELTA - ES))
FIN = (2.0*MUINF*(DELTA - ES)*FAC)/(R*(OMEGA + FAC))
FTS = TSG - FIN
FAC1 = R*(OMEGA+FAC)*(MUINF*(DELTA-ES)*(2.0/FAC)*LAMSQ*DES -
* 2.0*FAC*MUINF*DES) - 2.0*MUINF*(DELTA-ES)*R*LAMSQ*DES
FAC2 = R**2*(OMEGA + FAC)**2
DFTS = 1.0 - (FAC1/FAC2)
PS = FAC
IF (DFTS.EQ.0.0) THEN
  ICODE = 2
  RETURN
ENDIF
TSG1 = TSG - W*(FTS/DFTS)
DELTS = DABS(TSG1 - TSG)
TSG = TSG1
IF (ITER.GE.MAXI) THEN
  ICODE = 1
  RETURN
ENDIF
IF (DELTS.GT.TOL1) THEN
  ITER1 = ITER
  GO TO 2
ENDIF
ITER2 = ITER1 + 10
IF (DELTS.LE.TOL2.OR.ITER.GE.ITER2) RETURN
GO TO 2
END

```


STATEMENT LISTING FOR PROGRAM RAD.FOR

C PROGRAM FOR COMPUTING THE RADIATIVE HEAT TRANSFER AT THE BODY SURFACE
 C IN A HYPERSONIC FLOWFIELD. THIS PROGRAM IS USED IN CONJUNCTION WITH
 C AFE2.FOR

C RADIATION MODELS 1) OLSTAD MODEL
 C 2) CARLSON MODEL
 C

```

  C      IMPLICIT REAL*8(A-H,O-Z)
  C      COMMON /A9/ MUI(25),XPST(30)
  C      COMMON /RAD/ NXCON,NPTS(30),TSTAG,LI(25),IMAX
  C      COMMON /RAD1/ RRHO(30,30),Y(30,30),YBDY(30)
  C      COMMON /RAD2/ RCON(30,10,30)
  C      COMMON /RAD3/ B(30,8,30)
  C      COMMON /RAD5/ OPTL(30,8,30)
  C      COMMON /RAD6/ ALPN(30,30),ALPO(30,30),BETN(30,30),BETO(30,30)
  C      COMMON /RAD7/ TEMPO(30,30),TEMP(30,30)
  C      COMMON /RAD8/ KAP(30,4,30)
  C      COMMON /RAD9/ KAPPA(30,8,30),KA(30,4,30)
  C      COMMON /RAD10/ EPSIIL(20,25),GIL(20,25)
  C      REAL OPTL,XPST,Y,YBDY,TEMPO,TEMP,Z
  C      REAL ALPN,ALPO,B,BETN,BETO
  C      REAL*8 KAPPA,KAP,KA,MUI,EPSIIL,GIL
  C      CHARACTER*30 OUTFIL
  C
  C      OPEN RADIATION INPUT FILES (AFE2.FOR)
  C      OPEN (UNIT=15,FILE='RADIN1.DAT',STATUS='OLD')
  C      OPEN (UNIT=16,FILE='RADIN2.DAT',STATUS='OLD')
  C      OPEN (UNIT=17,FILE='RADIN3.DAT',STATUS='OLD')
  C
  C      READ INPUT
  C      READ(15,1) OUTFIL
  C      1 FORMAT(A30)
  C      READ(15,*) MODEL,IQ3,VINF,TINF,PINF
  C      READ(15,*) NXPST,NXCON,IMAX,IQ4,IQ5,IQ6,IQ8,IQ9
  C      IF (NXCON.GT.30) THEN
  C        WRITE(*,2) NXCON
  C      2  FORMAT(1X,'NXCON = ',I3,' MAX = 30 STRLINES IN RAD.FOR')
  C      STOP
  C      ENDIF
  C      IF (IQ4.EQ.0) GO TO 60
  C      OPEN (UNIT=8,FILE=OUTFIL,STATUS='UNKNOWN')
  C      DO 5 I = 1,IMAX
  C      5  READ(15,*) MUI(I)
  C      DO 7 I = 1,IMAX
  C      7  READ(15,*) LII
  C      LI(I) = LII
  C      DO 9 I = 1,IMAX
  C      9  READ(15,*) (GIL(L,I),L = 1,LII)
  C      DO 10 I = 1,NXPST
  C      10 READ(15,*) NPTS(I),XPST(I)
  C      READ(15,*) TSTAG
  C      READ(15,*) KSTAG
  C      DO 15 I = 1,KSTAG
  C      15 READ(16,*) LPS2,JJJ
  C      IF (IMAX.EQ.7) THEN
  C        READ(16,*) RCON(LPS2,1,JJJ),RCON(LPS2,2,JJJ),RCON(LPS2,3,JJJ),
  C        *      RCON(LPS2,4,JJJ),RCON(LPS2,5,JJJ),RCON(LPS2,6,JJJ),
  C        *      RCON(LPS2,7,JJJ),RRHO(LPS2,JJJ),TEMP(LPS2,JJJ),
  C        *      TEMPO(LPS2,JJJ)
  C      ENDIF
  C      ENDIF

```

```

      IF (IMAX.EQ.10) THEN
        READ(16,*) RCON(LPS2,1,JJJ),RCON(LPS2,2,JJJ),RCON(LPS2,3,JJJ),
        *      RCON(LPS2,4,JJJ),RCON(LPS2,5,JJJ),RCON(LPS2,6,JJJ),
        *      RCON(LPS2,7,JJJ),RCON(LPS2,8,JJJ),RCON(LPS2,9,JJJ),
        *      RCON(LPS2,10,JJJ),RRHO(LPS2,JJJ),TEMP(LPS2,JJJ),
        *      TEMPO(LPS2,JJJ)
      ENDIF
15  CONTINUE
19  READ(17,*,END=21) IBOB,IK
    READ(17,*) YBDY(IBOB)
    DO 20 J = 2,IK
20  READ(17,*) Y(IBOB,J-1)
    GO TO 19

C                                     INITIAL CONDITIONS
21  WRITE(*,22) VINP,PINF,TINF
    WRITE(8,22) VINP,PINF,TINF
22  FORMAT(1X,'FREESTREAM QUANTITIES : ',//,1X,'VINP = ',E12.5,1X,
    *      'cm/sec',/,1X,'PINF = ',E12.5,1X,'dynes/cm^2',/,1X,
    *      'TINF = ',E12.5,1X,'K'///)
    WRITE(*,23) MODEL
    WRITE(8,23) MODEL
23  FORMAT(1X,'VIBRATION-DISSOCIATION COUPLING MODELS',//,12X,
    *      'TYPE',11X,'NO.',//,9X,'VIB. EQUIL.',8X,'0',/,12X,'CVD',13X,
    *      '1',/,12X,'CVDV',12X,'2',/,6X,'CVDV-Preferential',5X,'3',/,
    *      12X,'PARK',12X,'4', ' - ',I3///)
    WRITE(*,24) IQ3
    WRITE(8,24) IQ3
24  FORMAT(7X,'SHOCK JUMP CONDITION MODELS',//,12X,
    *      'TYPE',11X,'NO.',//,6X,'CHEMISTRY FROZEN',6X,'1',/,5X,
    *      'N2 FROZEN, O2 DISS.',4X,'2',/,7X,'N2 AND O2 DISS.',6X,
    *      '3', ' - ',I3///)

C                                     RADIATION MODELS
      IF (IQ4.EQ.1) CALL OLRAD (IQ5,IQ6,IQ8,IQ9)
C      IF (IQ4.EQ.2) CALL CARLRAD (IQ4,IQ5,IQ9)
      IF ((IQ4.EQ.3).OR.(IQ4.EQ.4)) THEN
        CALL OLRAD (IQ5,IQ6,IQ8,IQ9)
C      CALL CARLRAD (IQ4,IQ5,IQ9)
      ENDIF
      IF (IQ4.EQ.4) THEN
        DO 40 K = 3,NXCON
          WRITE(*,30) XPST(K)
          WRITE(8,30) XPST(K)
30  FORMAT(///,'AT X = ',D11.4,///)
          DO 40 J = 1,NPTS(K)
            WRITE(*,35) J,TEMP(K,J),RRHO(K,J)
            WRITE(8,35) J,TEMP(K,J),RRHO(K,J)
35  FORMAT(' J = ',I2,' TVN2 = ',D11.4,' RHO = ',D11.4)
40  CONTINUE
          DO 55 K = 3,NXCON
            WRITE(*,30) XPST(K)
            WRITE(8,30) XPST(K)
            DO 55 J = 1,NPTS(K)
              WRITE(*,45) J
              WRITE(8,45) J
45  FORMAT(/,' AT J = ',I2,' THE NUMBER DENSITIES ARE ',/)
              DO 55 I = 1,IMAX
                WRITE(8,50) I,RCON(K,I,J)
                WRITE(*,50) I,RCON(K,I,J)
50  FORMAT(' SPECIES = ',I2,' N = ',D11.4)
55  CONTINUE

```

```

ENDIF
CLOSE(8)
60 CLOSE(15)
CLOSE(16)
CLOSE(17)
STOP
END

```

C
C-----
C

```

SUBROUTINE OLRAD (IQ5,IQ6,IQ8,IQ9)
IMPLICIT REAL*8(A-H,O-Z)
COMMON /A9/ MUI(25),XPST(30)
COMMON /RAD/ NXCON,NPTS(30),TSTAG,LI(25),IMAX
COMMON /RAD1/ RRHO(30,30),Y(30,30),YBDY(30)
COMMON /RAD2/ RCON(30,10,30)
COMMON /RAD3/ B(30,8,30)
COMMON /RAD5/ OPTL(30,8,30)
COMMON /RAD6/ ALPN(30,30),ALPO(30,30),BETN(30,30),BETO(30,30)
COMMON /RAD7/ TEMPO(30,30),TEMP(30,30)
COMMON /RAD8/ KAP(30,4,30)
COMMON /RAD9/ KAPPA(30,8,30),KA(30,4,30)
COMMON /RAD10/ EPSIIL(20,25),GIL(20,25)
REAL OPTL,XPST,Y,YBDY,TEMPO,TEMP,Z
REAL ALPN,ALPO,B,BETN,BETO,EPSIIL,GIL
REAL*8 K11,K22,N,NN,KAPPA,KAP,KA,MUI

```

C
C
C

```

WRITE(8,1)
1 FORMAT(///' OLSTAD RADIATION MODEL')
IF(IQ8.EQ.0) THEN
  IF(IQ6.EQ.0) WRITE(8,2)
  IF(IQ6.EQ.1) WRITE(8,3)
ENDIF
2 FORMAT('/' BETA=BETA(Te)')
3 FORMAT('/' BETA=BETA(Tt)')

```

C

NUMBER DENSITIES

```

DO 12 K=3,NXCON
DO 12 J=1,NPTS(K)
DO 10 II=1,IMAX
XXX=RRHO(K,J)*6.023D+23
XXX=XXX*RCON(K,II,J)/MUI(II)
10 RCON(K,II,J)=XXX
12 CONTINUE

```

C

NONEQUILIBRIUM RADIATION CORRECTION

```

IF (IQ5.EQ.0) THEN
DO 122 K=3,NXCON
DO 122 J=1,NPTS(K)
IF(J.EQ.NPTS(K)) THEN
  ALPN(K,J)=0.D+00
  BETN(K,J)=0.D+00
  GO TO 122
ENDIF
Q1=0.D+0
Q2=0.D+0
Q3=0.D+0
DO 112 III=1,LI(3)
112 Q1=Q1+GIL(III,3)*DEXP(-EPSIIL(III,3)/TEMP(K,J))
DO 113 III=1,LI(9)
113 Q2=Q2+GIL(III,9)*DEXP(-EPSIIL(III,9)/TEMP(K,J))
DO 130 III=1,LI(1)

```

```

130 Q3=Q3+GIL(III,1)*DEXP(-EPSIIL(III,1)/TEMP(K,J))
    RHO1=RCON(K,3,J)*1.401D+01/6.023D+23
    RHO2=RCON(K,9,J)*1.401D+01/6.023D+23
    RHO3=RCON(K,1,J)*2.802D+01/6.023D+23
    RHO4=RHO1+RHO2+RHO3
    QVIB=1.D+0/(1.D+0-DEXP(-3.39D+03/TEMP(K,J)))
    QROT=TEMP(K,J)/5.8D+0
    C1=RHO4*2.4701D+01*QVIB*QROT*Q3*DEXP(1.135D+05/TEMP(K,J))
    C1=C1/(Q1**2*TEMP(K,J)**1.5)
    C2=RHO4*8.909D+06*Q1*DEXP(1.69D+05/TEMP(K,J))
    C2=C2/(Q2*TEMP(K,J)**1.5)
    ALPHA=0.5D+0
    BETA=0.5D+0
132 F=BETA**2*(1.D+0-ALPHA)**2*C1+BETA-1.D+0
    FA=-2.D+0*BETA**2*(1.D+0-ALPHA)*C1
    FB=2.D+0*BETA*(1.D+0-ALPHA)**2*C1+1.D+0
    G=ALPHA**2*BETA*C2+ALPHA-1.D+0
    GA=2.D+0*ALPHA*BETA*C2+1.D+0
    GB=ALPHA**2*C2
    DENOM=FA*GB-FB*GA
    DA=(-F*GB+G*FB)/DENOM
    DB=(-G*FA+F*GA)/DENOM
    ALPHA=ALPHA+DA
    BETA=BETA+DB
    IF((DABS(DA).GT.1.D-05).OR.(DABS(DB).GT.1.D-5)) GO TO 132
    ALPH=RCON(K,9,J)/(RCON(K,9,J)+RCON(K,3,J))
    BET=(RCON(K,9,J)+RCON(K,3,J))/(2.D+0*RCON(K,1,J)+
*   RCON(K,9,J)+RCON(K,3,J))
    IF(IQ6.EQ.1) GO TO 1888
    BETN(K,J)=(BET**2/(1.D+0-BET))*((1.D+0-BETA)/(BETA**2))
*   *((1.D+0-ALPH)**2/(1.D+0-ALPHA)**2)
1888 ALPN(K,J)=BET*(ALPH**2/(1.D+0-ALPH))
*   *((1.D+0-ALPHA)/(BETA*ALPHA**2))
122 CONTINUE
    DO 142 K=3,NXCON
    DO 142 J=1,NPTS(K)
    IF(J.EQ.NPTS(K)) THEN
        ALPO(K,J)=0.D+00
        BETO(K,J)=0.D+00
        GO TO 142
    ENDIF
    Q1=0.D+0
    Q2=0.D+0
    Q3=0.D+0
    DO 152 III=1,LI(4)
152 Q1=Q1+GIL(III,4)*EXP(-EPSIIL(III,4)/TEMP(K,J))
    DO 153 III=1,LI(10)
153 Q2=Q2+GIL(III,10)*DEXP(-EPSIIL(III,10)/TEMP(K,J))
    DO 160 III=1,LI(2)
160 Q3=Q3+GIL(III,2)*DEXP(-EPSIIL(III,2)/TEMP(K,J))
    RHO1=RCON(K,4,J)*1.600D+01/6.023D+23
    RHO2=RCON(K,10,J)*1.600D+01/6.023D+23
    RHO3=RCON(K,2,J)*3.200D+01/6.023D+23
    RHO4=RHO1+RHO2+RHO3
    QVIB=1.D+0/(1.D+0-DEXP(-2.27D+03/TEMP(K,J)))
    QROT=TEMP(K,J)/4.2D+0
    C1=RHO4*1.7725D+01*QVIB*QROT*Q3*DEXP(5.950D+04/TEMP(K,J))
    C1=C1/(Q1**2*TEMP(K,J)**1.5)
    C2=RHO4*7.801D+06*Q1*DEXP(1.58D+05/TEMP(K,J))
    C2=C2/(Q2*TEMP(K,J)**1.5)

```

```

      ALPHA=0.5D+0
      BETA=0.5D+0
162  F=BETA**2*(1.D+0-ALPHA)**2*C1+BETA-1.D+0
      FA=-2.D+0*BETA**2*(1.D+0-ALPHA)*C1
      FB=2.D+0*BETA*(1.D+0-ALPHA)**2*C1+1.D+0
      G=ALPHA**2*BETA*C2+ALPHA-1.D+0
      GA=2.D+0*ALPHA*BETA*C2+1.D+0
      GB=ALPHA**2*C2
      DENOM=FA*GB-FB*GA
      DA=(-F*GB+G*FB)/DENOM
      DB=(-G*FA+F*GA)/DENOM
      ALPHA=ALPHA+DA
      BETA=BETA+DB
      IF((DABS(DA).GT.1.D-05).OR.(DABS(DB).GT.1.D-5)) GO TO 162
      ALPH=RCON(K,10,J)/(RCON(K,10,J)+RCON(K,4,J))
      BET=(RCON(K,10,J)+RCON(K,4,J))/(2.D+0*RCON(K,2,J)+
*      RCON(K,10,J)+RCON(K,4,J))
      IF(IQ6.EQ.1) GO TO 1999
      BETO(K,J)=(BET**2/(1.D+0-BET))*((1.D+0-BETA)/(BETA**2))
*      *((1.D+0-ALPH)**2/(1.D+0-ALPHA)**2)
1999  ALPO(K,J)=BET*(ALPH**2/(1.D+0-ALPH))
*      *((1.D+0-ALPHA)/(BETA*ALPHA**2))
142  CONTINUE
C
      BETA=BETA(Tt)
      IF(IQ6.EQ.1) THEN
        DO 1122 K=3,NXCON
        DO 1122 J=1,NPTS(K)
        IF(J.EQ.NPTS(K)) THEN
          BETN(K,J)=0.D+00
          GO TO 1122
        ENDIF
        Q1=0.D+0
        Q2=0.D+0
        Q3=0.D+0
        DO 1112 III=1,LI(3)
1112  Q1=Q1+GIL(III,3)*DEXP(-EPSIIL(III,3)/TEMPO(K,J))
        DO 1113 III=1,LI(9)
1113  Q2=Q2+GIL(III,9)*DEXP(-EPSIIL(III,9)/TEMPO(K,J))
        DO 1130 III=1,LI(1)
1130  Q3=Q3+GIL(III,1)*DEXP(-EPSIIL(III,1)/TEMPO(K,J))
        RHO1=RCON(K,3,J)*1.401D+01/6.023D+23
        RHO2=RCON(K,9,J)*1.401D+01/6.023D+23
        RHO3=RCON(K,1,J)*2.802D+01/6.023D+23
        RHO4=RHO1+RHO2+RHO3
        QVIB=1.D+0/(1.D+0-DEXP(-3.39D+03/TEMPO(K,J)))
        QROT=TEMPO(K,J)/5.8D+0
        C1=RHO4*2.4701D+01*QVIB*QROT*Q3*DEXP(1.135D+05/TEMPO(K,J))
        C1=C1/(Q1**2*TEMPO(K,J)**1.5)
        C2=RHO4*8.909D+06*Q1*DEXP(1.69D+05/TEMPO(K,J))
        C2=C2/(Q2*TEMPO(K,J)**1.5)
        ALPHA=0.5D+0
        BETA=0.5D+0
1132  F=BETA**2*(1.D+0-ALPHA)**2*C1+BETA-1.D+0
        FA=-2.D+0*BETA**2*(1.D+0-ALPHA)*C1
        FB=2.D+0*BETA*(1.D+0-ALPHA)**2*C1+1.D+0
        G=ALPHA**2*BETA*C2+ALPHA-1.D+0
        GA=2.D+0*ALPHA*BETA*C2+1.D+0
        GB=ALPHA**2*C2
        DENOM=FA*GB-FB*GA
        DA=(-F*GB+G*FB)/DENOM

```

```

      DB=(-G*FA+F*GA)/DENOM
      ALPHA=ALPHA+DA
      BETA=BETA+DB
      IF((DABS(DA).GT.1.D-05).OR.(DABS(DB).GT.1.D-5)) GO TO 1132
      BET=(RCON(K,9,J)+RCON(K,3,J))/(2.D+0*RCON(K,1,J)+
*      RCON(K,9,J)+RCON(K,3,J))
      ALPH=RCON(K,9,J)/(RCON(K,9,J)+RCON(K,3,J))
      BETN(K,J)=(BET**2/(1.D+0-BET))*((1.D+0-BETA)/(BETA**2))
*      *((1.D+0-ALPH)**2/(1.D+0-ALPHA)**2)
1122 CONTINUE
      DO 1142 K=3,NXCON
      DO 1142 J=1,NPTS(K)
      IF(J.EQ.NPTS(K)) THEN
        BETO(K,J)=0.D+00
        GO TO 1142
      ENDIF
      Q1=0.D+0
      Q2=0.D+0
      Q3=0.D+0
      DO 1152 III=1,LI(4)
1152 Q1=Q1+GIL(III,4)*EXP(-EPSIIL(III,4)/TEMPO(K,J))
      DO 1153 III=1,LI(10)
1153 Q2=Q2+GIL(III,10)*DEXP(-EPSIIL(III,10)/TEMPO(K,J))
      DO 1160 III=1,LI(2)
1160 Q3=Q3+GIL(III,2)*DEXP(-EPSIIL(III,2)/TEMPO(K,J))
      RHO1=RCON(K,4,J)*1.600D+01/6.023D+23
      RHO2=RCON(K,10,J)*1.600D+01/6.023D+23
      RHO3=RCON(K,2,J)*3.200D+01/6.023D+23
      RHO4=RHO1+RHO2+RHO3
      QVIB=1.D+0/(1.D+0-DEXP(-2.27D+03/TEMPO(K,J)))
      QROT=TEMPO(K,J)/4.2D+0
      C1=RHO4*1.7725D+01*QVIB*QROT*Q3*DEXP(5.950D+04/TEMPO(K,J))
      C1=C1/(Q1**2*TEMPO(K,J)**1.5)
      C2=RHO4*7.801D+06*Q1*DEXP(1.58D+05/TEMPO(K,J))
      C2=C2/(Q2*TEMPO(K,J)**1.5)
      ALPHA=0.5D+0
      BETA=0.5D+0
1162 F=BETA**2*(1.D+0-ALPHA)**2*C1+BETA-1.D+0
      FA=-2.D+0*BETA**2*(1.D+0-ALPHA)*C1
      FB=2.D+0*BETA*(1.D+0-ALPHA)**2*C1+1.D+0
      G=ALPHA**2*BETA*C2+ALPHA-1.D+0
      GA=2.D+0*ALPHA*BETA*C2+1.D+0
      GB=ALPHA**2*C2
      DENOM=FA*GB-FB*GA
      DA=(-F*GB+G*FB)/DENOM
      DB=(-G*FA+F*GA)/DENOM
      ALPHA=ALPHA+DA
      BETA=BETA+DB
      IF((DABS(DA).GT.1.D-05).OR.(DABS(DB).GT.1.D-5)) GO TO 1162
      BET=(RCON(K,10,J)+RCON(K,4,J))/(2.D+0*RCON(K,2,J)+
*      RCON(K,10,J)+RCON(K,4,J))
      ALPH=RCON(K,10,J)/(RCON(K,10,J)+RCON(K,4,J))
      BETO(K,J)=(BET**2/(1.D+0-BET))*((1.D+0-BETA)/(BETA**2))
*      *((1.D+0-ALPH)**2/(1.D+0-ALPHA)**2)
1142 CONTINUE
      ENDIF
      ENDIF
C      ABSORPTION COEFFICIENTS
      DO 20 K=3,NXCON
      DO 20 J=1,NPTS(K)

```

```

TT=TEMP(K,J)/168800.0
IF(IQ6.EQ.1) THEN
    TM=TEMPO(K,J)/168800.0
ELSE
    TM=TT
ENDIF
C IF((IQ6.EQ.1).AND.(TEMPO(K,J).LE.8.D+03)) GO TO 20
C IF(TEMP(K,J).LE.8.D+03) GO TO 20
IF(IQ8.EQ.0) THEN
    O=BETO(K,J)
    N=BETN(K,J)
ELSE
    O=1.D+0
    N=1.D+0
ENDIF
KAPPA(K,4,J)=5.D-19*O*RCON(K,2,J)+5.D-20*N*RCON(K,1,J)
KAP(K,4,J)=1.7D-17*RCON(K,3,J)*DEXP(-.246/TT)
KA(K,4,J)=0.D+0
KAPPA(K,3,J)=2.D-18*(N*RCON(K,1,J)+O*RCON(K,2,J))+KAPPA(K,4,J)
KAP(K,3,J)=2.1D-17*RCON(K,3,J)*DEXP(-.165/TT)+KAP(K,4,J)
KA(K,3,J)=0.D+0
KAPPA(K,2,J)=5.1D-18*(N*RCON(K,1,J)+O*RCON(K,2,J))+KAPPA(K,3,J)
KAP(K,2,J)=KAP(K,3,J)
KA(K,2,J)=5.1D-18*RCON(K,4,J)
KAPPA(K,1,J)=2.D-17*O*RCON(K,2,J)+4.D-16*N*RCON(K,1,J)
C      +KAPPA(K,2,J)
KAP(K,1,J)=1.1D-17*RCON(K,3,J)+KAP(K,2,J)
KA(K,1,J)=KA(K,2,J)
IF(IQ5.EQ.0) THEN
    N=ALPN(K,J)
    O=ALPO(K,J)
    IF(IQ8.EQ.0) THEN
        NN=BETN(K,J)
        OO=BETO(K,J)
    ELSE
        NN=1.D+0
        OO=1.D+0
    ENDIF
    KAPPA(K,5,J)=7.7D-17*(NN*RCON(K,1,J)+OO*RCON(K,2,J))
C      *DEXP(-.49/TT)+
C      2.6D-17*(RCON(K,3,J)*N+RCON(K,4,J)*O)*DEXP(-.723/TT)
    KAPPA(K,6,J)=2.D-18*RCON(K,2,J)*OO+1.5D-17*(RCON(K,3,J)*N+
C      RCON(K,4,J)*O)*DEXP(-.379/TT)+KAPPA(K,5,J)
    IF(RCON(K,7,J).LE.1.D+00) THEN
        KAPPA(K,7,J)=KAPPA(K,6,J)
        GO TO 909
    ENDIF
    KAPPA(K,7,J)=3.D+03*((RCON(K,3,J)*N+RCON(K,4,J)*O)/RCON(K,7,J))*
C      DEXP(-.489/TT)+KAPPA(K,6,J)
909 KAPPA(K,8,J)=3.2D-17*(RCON(K,3,J)*N+RCON(K,4,J)*O)*DEXP(-.631/TT)
C      +KAPPA(K,5,J)
    ELSE
        KAPPA(K,5,J)=7.7D-17*(RCON(K,1,J)+RCON(K,2,J))*DEXP(-.49/TT)+
C      2.6D-17*(RCON(K,3,J)+RCON(K,4,J))*DEXP(-.723/TT)
        KAPPA(K,6,J)=2.D-18*RCON(K,2,J)+1.5D-17*(RCON(K,3,J)+RCON(K,4,J))
C      *DEXP(-.379/TT)+KAPPA(K,5,J)
        IF(RCON(K,7,J).LE.1.D+00) THEN
            KAPPA(K,7,J)=KAPPA(K,6,J)
            GO TO 910
        ENDIF

```



```

      KAPPA(K,7,J)=3.D+03*((RCON(K,3,J)+RCON(K,4,J))/RCON(K,7,J))*
C      DEXP(-.489/TT)+KAPPA(K,6,J)
910 KAPPA(K,8,J)=3.2D-17*(RCON(K,3,J)+RCON(K,4,J))*DEXP(-.631/TT)
C      +KAPPA(K,5,J)

```

```

      ENDIF

```

```

20 CONTINUE

```

PLANCK FUNCTION

```

C      DO 30 K=3,NXCON
      DO 30 J=1,NPTS(K)
      TT=TEMP(K,J)/168800.0
C      IF(TEMP(K,J).LE.8.D+03) GO TO 30
      BLACK=(TT*168800.)*4*5.6696D-12/3.1415927
      FF=15.D+00/(3.1415927*4)
      X1=1./TT
      B1=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C      DEXP(-2.*X1)*0.5*(.75+1.5*X1+1.5*X1**2+X1**3))
      X1=0.935/TT
      B2=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C      DEXP(-2.*X1)*0.5*(.75+1.5*X1+1.5*X1**2+X1**3))
      X1=0.835/TT
      B3=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C      DEXP(-2.*X1)*0.5*(.75+1.5*X1+1.5*X1**2+X1**3))
      X1=0.754/TT
      B4=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C      DEXP(-2.*X1)*0.5*(.75+1.5*X1+1.5*X1**2+X1**3))
      X1=0.473/TT
      B6=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C      DEXP(-2.*X1)*0.5*(.75+1.5*X1+1.5*X1**2+X1**3))
      X1=0.213/TT
      B8=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C      DEXP(-2.*X1)*0.5*(.75+1.5*X1+1.5*X1**2+X1**3))
      B8=1-B8
      B2=B2-B1
      B3=B3-B2
      B4=B4-B3
      B6=B6-B4
      B5=1-B8-B6-B4-B3-B2-B1
      B(K,1,J)=BLACK*B1
      B(K,2,J)=BLACK*B2
      B(K,3,J)=BLACK*B3
      B(K,4,J)=BLACK*B4
      B(K,6,J)=BLACK*B6
      B(K,7,J)=BLACK*2.4D-21*RCON(K,7,J)*B6*DEXP(.162/TT)
      B(K,8,J)=BLACK*B8
      B(K,5,J)=BLACK*B5

```

```

30 CONTINUE

```

TAU'S

```

C      DO 100 K=3,NXCON
      DO 100 I=1,8
      IF(I.GT.4) THEN
      OPTL(K,I,1)=KAPPA(K,I,1)*DABS(Y(K,1)-YBDY(K))
      ELSE
      OPTL(K,I,1)=(KAPPA(K,I,1)+KAP(K,I,1)+KA(K,I,1))
C      *DABS(Y(K,1)-YBDY(K))
      ENDIF
      DO 100 J=2,NPTS(K)
      IF(I.GT.4) THEN
      K11=KAPPA(K,I,J)+KAPPA(K,I,J-1)
      ELSE
      K11=KAPPA(K,I,J)+KAP(K,I,J)+KAPPA(K,I,J-1)+KAP(K,I,J-1)

```

```

      C      +KA(K,I,J)+KA(K,I,J-1)
      ENDIF
      OPTL(K,I,J)=OPTL(K,I,J-1)+(5.D-1*(K11)*DABS(Y(K,J-1)
      C      -Y(K,J)))
100 CONTINUE
C
      QRW
      DO 200 K=3,NXCON
      WRITE(*,102) XPST(K)
      WRITE(8,102) XPST(K)
102 FORMAT(///,' AT X = ',D11.4)
      WQR=0.0
      DO 180 I=1,8
      CALL FEI(E1I,E2I,E3I,OPTL(K,I,1))
      EOLD=E2I
      JIN=1
      IF(I.GT.4) THEN
        K11=KAPPA(K,I,1)*B(K,I,1)
      ELSE
        IF(IQ5.EQ.1) THEN
          ALPN(K,1)=1.D+0
          ALPO(K,1)=1.D+0
        ENDIF
        K11=KAPPA(K,I,1)*B(K,I,1)+KAP(K,I,1)*B(K,I,1)*ALPN(K,1)+
      C      KA(K,I,1)*B(K,I,1)*ALPO(K,1)
      ENDIF
      SUM=3.1415927*EOLD*K11*DABS(Y(K,1)-YBDY(K))
      IF(IQ9.EQ.0) WRITE(8,2221) JIN,I,SUM
      DO 150 J=2,NPTS(K)
      IF(I.GT.4) THEN
        K11=KAPPA(K,I,J)*B(K,I,J)
        K22=KAPPA(K,I,J-1)*B(K,I,J-1)
      ELSE
        IF(IQ5.EQ.1) THEN
          ALPN(K,J)=1.D+0
          ALPO(K,J)=1.D+0
        ENDIF
        K11=KAPPA(K,I,J)*B(K,I,J)+KAP(K,I,J)*B(K,I,J)*ALPN(K,J)+
      C      KA(K,I,J)*B(K,I,J)*ALPO(K,J)
        K22=KAPPA(K,I,J-1)*B(K,I,J-1)+
      C      KAP(K,I,J-1)*B(K,I,J-1)*ALPN(K,J-1)+
      C      KA(K,I,J-1)*B(K,I,J-1)*ALPO(K,J-1)
      ENDIF
      Z=OPTL(K,I,J)
      CALL FEI(E1I,E2I,E3I,Z)
      STSUM=3.14159*(K11*E2I+K22*EOLD)*DABS(Y(K,J-1)-Y(K,J))
      IF(IQ9.EQ.0) THEN
        WRITE(8,2221) J,I,STSUM
2221 FORMAT(1X,' AT J = ',I2,' BAND ',I2,' QR = ',D11.4)
      ENDIF
145 SUM=SUM+STSUM
      EOLD=E2I
150 CONTINUE
      WRITE(*,103) I,SUM
      WRITE(8,103) I,SUM
103 FORMAT(/,' FOR BAND ',I2,' QR = ',D11.4,' WATTS/SQ.CM.')
      WQR=WQR+SUM
180 CONTINUE
      WRITE(*,101) WQR
      WRITE(8,101) WQR
101 FORMAT(/' TOTAL QR = ',D11.4,' WATTS/SQ.CM.')

```

200 CONTINUE

RETURN

END

C

C-----

C

```
SUBROUTINE FEI(E1I,E2I,E3I,Z)
IMPLICIT REAL*8(A-H,O-Z)
COMMON /RAD1/ RRHO(30,30),Y(30,30),YBDY(30)
COMMON /RAD3/ B(30,8,30)
COMMON /RAD5/ OPTL(30,8,30)
C REAL OPTL,XPST,Y,YBDY,TEMPO,TEMP,Z
C REAL ALPN,ALPO,B,BETN,BETO
CALL EXPI(Z,E1I,AUX)
E2I=DEXP(-Z)-Z*E1I
E3I=(DEXP(-Z)-Z*E2I)/2.0
C E2I=DEXP(-Z*DSQRT(3.D+00))
RETURN
END
```

C

C-----

C

```
SUBROUTINE EXPI(X,RES,AUX)
IMPLICIT REAL*8(A-H,O-Z)
COMMON /RAD1/ RRHO(30,30),Y(30,30),YBDY(30)
COMMON /RAD3/ B(30,8,30)
COMMON /RAD5/ OPTL(30,8,30)
C REAL OPTL,XPST,Y,YBDY,TEMPO,TEMP,Z
C REAL ALPN,ALPO,B,BETN,BETO
IF(X-1.) 2,1,1
1 YY=1./X
AUX=1.-YY*(((YY+3.377358)*YY+2.052156)*YY+2.709479D-01)/(((YY*
C1.072553+5.716943)*YY+6.945239)*YY+2.593888)*YY+2.709496D-01)
RES=AUX*YY*DEXP(-X)
RETURN
2 IF(X+3.) 6,6,3
3 AUX=((((((7.122452D-7*X-1.766345D-6)*X+2.928433D-5)*X-2.335379D-4
C)*X+1.664156D-3)*X-1.041576D-2)*X+5.555682D-2)*X-2.500001D-1)*X
C+9.999999D-1
RES=-1.D+30
IF(X) 4,5,4
4 RES=X*AUX-DLOG(ABS(X))-5.772157D-1
5 RETURN
6 IF(X+9.) 8,8,7
7 AUX=1.-((((5.176245D-2*X+3.061037)*X+3.243655D+1)*X+2.244234D+2)*X
C+2.486697D+2)/((((X+3.995161)*X+3.893944D+1)*X+2.263818D+1)*X
C+1.807837D+2)
GO TO 9
8 YY=9./X
AUX=1.-YY*(((YY+7.659824D-1)*YY-7.271015D-1)*YY-1.080693)/(((YY
C*2.518750+1.122927D+1)*YY+5.921405)*YY-8.666702)*YY-9.724216)
9 RES=AUX*DEXP(-X)/X
RETURN
END
```

LISTING OF INPUT FILE RR3.NUM

[illegible]

5.675E-04,5.675E-04,5.675E-04,5.675E-04,5.675E-04,5.675E-04,5.675E-04,5.675E-04,
5.675E-04,5.675E-04
0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0
-.09952,-.09952,-.09952,-.09952,-.09952,-.09952,-.09952,-.09952,-.09952
.37177,.37177,.37177,.37177,.37177,.37177,.37177,.37177
0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0
-.2226,-.2226,-.2226,-.2226,-.2226,-.2226,-.2226,-.2226,-.2226,-.2226
-.2226,-.2226,-.2226,-.2226,-.2226,-.2226,-.2226,-.2226,-.2226,-.2226
0,0,0,0,0,0,0,0,0,0
-.09952,-.09952,-.09952,-.09952,-.09952,-.09952,-.09952,-.09952,-.09952,-.09952
0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0
163.63,163.63,163.63,163.63,163.63,163.63,163.63,163.63,163.63,163.63
179.32,179.32,179.32,179.32,179.32,179.32,179.32,179.32,179.32,179.32
0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0
135.3,135.3,135.3,135.3,135.3,135.3,135.3,135.3,135.3,135.3
135.3,135.3,135.3,135.3,135.3,135.3,135.3,135.3,135.3,135.3
0,0,0,0,0,0,0,0,0,0
163.63,163.63,163.63,163.63,163.63,163.63,163.63,163.63,163.63,163.63
0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0
1,3,6,2
3,2,1,3,1,3
4,6,4,6
5,3,1,5,1
2,2,2,4,2,4,2,4
1,6,3
1
2,4,2,2
1,3,5,5,1,5
4,6,4,4,2
0,71592,85343,99212
0,11342,18879,51385,52104,71026
0,27659,27670,41496
0,228,326,22831,48622
0,174,63597,65381,76676,75361,87568,86359
0,57528,84205
0
0,13191,36634,92977.4
0,70.5,188.5,22036.5,47029.68,67864.8
0,38571,38601,58223.5,58225
46,18800,3393.9685,20.797195,1.0801997E-02,-7.32343E-04
33,9896,2273.1279,17.365295,7.8529954E-02,-2.057468E-03
0,0,0,0,0,0
0,0,0,0,0,0
0,0,0,0,0,0
0,0,0,0,0,0
0,0,0,0,0,0
0,0,0,0,0,0
0,0,0,0,0,0
0,0,0,0,0,0
0,0,0,0,0,0
0,0,0,0,0,0
N2
O2
N
O
NO
NO+

E-
 N2+
 N+
 O+
 O2 + M..>2O + M
 2O + M..>O2 + M
 NO + M..>N + O + M
 N + O + M..>NO + M
 N2 + M..>2N + M
 2N + M..>N2 + M
 N + O2..>NO + O
 NO + O..>N + O2
 N2 + O..>NO + N
 NO + N..>N2 + O
 NO+ + E-..>N + O
 N + O..>NO+ + E-
 N + N..>N2+ + E-
 N2+ + E-..>N + N
 N + N..>N+ + E- + N
 N+ + E- + N..>N + N
 N + N+..>N+ + E- + N+
 N+ + E- + N+..>N+ + N+
 N + E-..>N+ + 2E-
 N+ + 2E-..>N + E-
 O + M..>O+ + E- + M
 O+ + E- + M..>O + M

STATEMENT LISTING FOR PROGRAM RADMOD.FOR

```

C PROGRAM FOR COMPUTING THE RADIATIVE HEAT TRANSFER AT THE BODY SURFACE
C IN A HYPERSONIC FLOWFIELD. THIS PROGRAM IS USED IN CONJUNCTION WITH
C AFE2.FOR
C
C RADIATION MODELS                1) OLSTAD MODEL
C                                2) CARLSON MODEL
C
C      IMPLICIT REAL*8(A-H,O-Z)
C      COMMON /A9/ MUI(25),XPST(30)
C      COMMON /RAD/ NXCON,NPTS(30),TSTAG,LI(25),IMAX
C      COMMON /RAD1/ RRHO(30,30),Y(30,30),YBDY(30)
C      COMMON /RAD2/ RCON(30,10,30)
C      COMMON /RAD3/ B(30,8,30)
C      COMMON /RAD5/ OPTL(30,8,30)
C      COMMON /RAD6/ ALPN(30,30),ALPO(30,30),BETN(30,30),BETO(30,30)
C      COMMON /RAD7/ TEMPO(30,30),TEMP(30,30)
C      COMMON /RAD8/ KAP(30,4,30)
C      COMMON /RAD9/ KAPPA(30,8,30),KA(30,4,30)
C      COMMON /RAD10/ EPSIIL(20,25),GIL(20,25)
C      REAL OPTL,XPST,Y,YBDY,TEMPO,TEMP,Z
C      REAL ALPN,ALPO,B,BETN,BETO
C      REAL*8 KAPPA,KAP,KA,MUI,EPSIIL,GIL
C      CHARACTER*30 OUTFIL
C
C                                OPEN RADIATION INPUT FILES (AFE2.FOR)
C      OPEN (UNIT=15,FILE='RADIN1.DAT',STATUS='OLD')
C      OPEN (UNIT=16,FILE='RADIN2.DAT',STATUS='OLD')
C      OPEN (UNIT=17,FILE='RADIN3.DAT',STATUS='OLD')
C
C                                READ INPUT
C      READ(15,1) OUTFIL
C      1 FORMAT(A30)
C      READ(15,*) MODEL,IQ3,VINF,TINF,PINF
C      READ(15,*) NXCON,NXPST,NXCON,IMAX,IQ4,IQ5,IQ6,IQ8,IQ9
C      IF (NXCON.GT.30) THEN
C        WRITE(*,2) NXCON
C      2  FORMAT(1X,'NXCON = ',I3,' MAX = 30 STRLINES IN RAD.FOR')
C      STOP
C      ENDIF
C      IF (IQ4.EQ.0) GO TO 60
C      OPEN (UNIT=8,FILE=OUTFIL,STATUS='UNKNOWN')
C      DO 5 I = 1,IMAX
C      5  READ(15,*) MUI(I)
C      DO 7 I = 1,IMAX
C      7  READ(15,*) LII
C      LI(I) = LII
C      7  READ(15,*) (GIL(L,I),L = 1,LII)
C      DO 9 I = 1,IMAX
C      9  READ(15,*) (EPSIIL(L,I),L = 1,LII)
C      DO 10 I = 1,NXPST
C      10 READ(15,*) NPTS(I),XPST(I)
C      READ(15,*) TSTAG
C      READ(15,*) KSTAG
C      DO 15 I = 1,KSTAG
C      15 READ(16,*) LPS2,JJJ
C      IF (IMAX.EQ.7) THEN
C        READ(16,*) RCON(LPS2,1,JJJ),RCON(LPS2,2,JJJ),RCON(LPS2,3,JJJ),
C        *      RCON(LPS2,4,JJJ),RCON(LPS2,5,JJJ),RCON(LPS2,6,JJJ),
C        *      RCON(LPS2,7,JJJ),RRHO(LPS2,JJJ),TEMP(LPS2,JJJ),
C        *      TEMPO(LPS2,JJJ)
C      ENDIF

```

```

      IF (IMAX.EQ.10) THEN
        READ(16,*) RCON(LPS2,1,JJJ),RCON(LPS2,2,JJJ),RCON(LPS2,3,JJJ),
        *      RCON(LPS2,4,JJJ),RCON(LPS2,5,JJJ),RCON(LPS2,6,JJJ),
        *      RCON(LPS2,7,JJJ),RCON(LPS2,8,JJJ),RCON(LPS2,9,JJJ),
        *      RCON(LPS2,10,JJJ),RRHO(LPS2,JJJ),TEMP(LPS2,JJJ),
        *      TEMPO(LPS2,JJJ)
      ENDIF
15  CONTINUE
19  READ(17,*,END=21) IBOB,IK
    READ(17,*) YBDY(IBOB)
    DO 20 J = 2,IK
20  READ(17,*) Y(IBOB,J-1)
    GO TO 19

C                                     INITIAL CONDITIONS
21  WRITE(*,22) VINFINF,PINF,TINF
    WRITE(8,22) VINFINF,PINF,TINF
22  FORMAT(1X,'FREESTREAM QUANTITIES : ',//,1X,'VINFINF = ',E12.5,1X,
    *      'cm/sec',/,1X,'PINF = ',E12.5,1X,'dynes/cm^2',/,1X,
    *      'TINF = ',E12.5,1X,'K'//)
    WRITE(*,23) MODEL
    WRITE(8,23) MODEL
23  FORMAT(1X,'VIBRATION-DISSOCIATION COUPLING MODELS',//,12X,
    *      'TYPE',11X,'NO.',//,9X,'VIB. EQUIL.',8X,'0',/,12X,'CVD',13X,
    *      '1',/,12X,'CVDV',12X,'2',/,6X,'CVDV-Preferential',5X,'3',/,
    *      12X,'PARK',12X,'4', ' = ',I3//)
    WRITE(*,24) IQ3
    WRITE(8,24) IQ3
24  FORMAT(7X,'SHOCK JUMP CONDITION MODELS',//,12X,
    *      'TYPE',11X,'NO.',//,6X,'CHEMISTRY FROZEN',6X,'1',/,5X,
    *      'N2 FROZEN, O2 DISS.',4X,'2',/,7X,'N2 AND O2 DISS.',6X,
    *      '3', ' = ',I3//)

C                                     RADIATION MODELS
    IF (IQ4.EQ.1) CALL OLRAD (IQ5,IQ6,IQ8,IQ9)
C    IF (IQ4.EQ.2) CALL CARLRAD (IQ4,IQ5,IQ9)
    IF ((IQ4.EQ.3).OR.(IQ4.EQ.4)) THEN
      CALL OLRAD (IQ5,IQ6,IQ8,IQ9)
C    CALL CARLRAD (IQ4,IQ5,IQ9)
    ENDIF
    IF (IQ4.EQ.4) THEN
      IF (IQ5.EQ.0) THEN
        DO 1920 K=3, NXCON
          WRITE(8,30)XPST(K)
          DO 1920 J=1,NPTS(K)
            WRITE(8,1921)J,ALPN(K,J),ALPO(K,J),BETN(K,J),BETO(K,J)
1921  FORMAT(' J=',I2,' An=',D11.4,' Ao=',D11.4/
            *      ',2X,' BN=',D11.4,' AO=',D11.4)
1920  CONTINUE
          ENDIF
          DO 40 K = 3,NXCON
            WRITE(*,30) XPST(K)
            WRITE(8,30) XPST(K)
30  FORMAT(///,'AT X = ',D11.4,///)
            DO 40 J = 1,NPTS(K)
              WRITE(*,35) J,TEMP(K,J),RRHO(K,J),TEMPO(K,J)
              WRITE(8,35) J,TEMP(K,J),RRHO(K,J),TEMPO(K,J)
35  FORMAT(' J = ',I2,' TVN2 = ',F10.2,' RHO = ',D11.4,' T=',F10.2)
40  CONTINUE
            DO 55 K = 3,NXCON
              WRITE(*,30) XPST(K)
              WRITE(8,30) XPST(K)

```

```

      DO 55 J = 1,NPTS(K)
      WRITE(*,45) J
      WRITE(8,45) J
45  FORMAT(/,' AT J = ',I2,' THE NUMBER DENSITIES ARE ',/)
      DO 55 I = 1,IMAX
      WRITE(8,50) I,RCON(K,I,J)
      WRITE(*,50) I,RCON(K,I,J)
50  FORMAT(' SPECIES = ',I2,' N = ',D11.4)
55  CONTINUE
      ENDIF
      CLOSE(8)
60  CLOSE(15)
      CLOSE(16)
      CLOSE(17)
      STOP
      END

```

C

C-----

C

```

      SUBROUTINE OLRAD (IQ5,IQ6,IQ8,IQ9)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /A9/ MUI(25),XPST(30)
      COMMON /RAD/ NXCON,NPTS(30),TSTAG,LI(25),IMAX
      COMMON /RAD1/ RRHO(30,30),Y(30,30),YBDY(30)
      COMMON /RAD2/ RCON(30,10,30)
      COMMON /RAD3/ B(30,8,30)
      COMMON /RAD5/ OPTL(30,8,30)
      COMMON /RAD6/ ALPN(30,30),ALPO(30,30),BETN(30,30),BETO(30,30)
      COMMON /RAD7/ TEMPO(30,30),TEMP(30,30)
      COMMON /RAD8/ KAP(30,4,30)
      COMMON /RAD9/ KAPPA(30,8,30),KA(30,4,30)
      COMMON /RAD10/ EPSIIL(20,25),GIL(20,25)
C     REAL OPTL,XPST,Y,YBDY,TEMPO,TEMP,Z
C     REAL ALPN,ALPO,B,BETN,BETO,EPSIIL,GIL
      REAL*8 K11,K22,N,NN,KAPPA,KAP,KA,MUI

```

C

C

C

```

      WRITE(8,1)
1  FORMAT(///' OLRAD RADIATION MODEL')
      IF(IQ8.EQ.0) THEN
      IF(IQ6.EQ.0) WRITE(8,2)
      IF(IQ6.EQ.1) WRITE(8,3)
      ENDIF
2  FORMAT('/ BETA=BETA(Te)')
3  FORMAT('/ BETA=BETA(Tt)')

```

C

NUMBER DENSITIES

```

      DO 12 K=3,NXCON
      DO 12 J=1,NPTS(K)
      DO 10 II=1,IMAX
      XXX=RRHO(K,J)*6.023D+23
      XXX=XXX*RCON(K,II,J)/MUI(II)
10  RCON(K,II,J)=XXX
12  CONTINUE

```

C

NONEQUILIBRIUM RADIATION CORRECTION

```

      IF (IQ5.EQ.0) THEN
      DO 122 K=3,NXCON
      DO 122 J=1,NPTS(K)
      IF(J.EQ.NPTS(K)) THEN
      ALPN(K,J)=0.D+00
      BETN(K,J)=0.D+00
      GO TO 122

```

```

ENDIF
Q1=0.D+0
Q2=0.D+0
Q3=0.D+0
DO 112 III=1,LI(3)
112 Q1=Q1+GIL(III,3)*DEXP(-EPSIIL(III,3)/TEMP(K,J))
DO 113 III=1,LI(9)
113 Q2=Q2+GIL(III,9)*DEXP(-EPSIIL(III,9)/TEMP(K,J))
DO 130 III=1,LI(1)
130 Q3=Q3+GIL(III,1)*DEXP(-EPSIIL(III,1)/TEMP(K,J))
RHO1=RCON(K,3,J)*1.401D+01/6.023D+23
RHO2=RCON(K,9,J)*1.401D+01/6.023D+23
RHO3=RCON(K,1,J)*2.802D+01/6.023D+23
RHO4=RHO1+RHO2+RHO3
QVIB=1.D+0/(1.D+0-DEXP(-3.39D+03/TEMP(K,J)))
QROT=TEMP(K,J)/5.8D+0
C1=RHO4*2.4701D+01*QVIB*QROT*Q3*DEXP(1.135D+05/TEMP(K,J))
C1=C1/(Q1**2*TEMP(K,J)**1.5)
C2=RHO4*8.909D+06*Q1*DEXP(1.69D+05/TEMP(K,J))
C2=C2/(Q2*TEMP(K,J)**1.5)
ALPHA=0.5D+0
BETA=0.5D+0
132 F=BETA**2*(1.D+0-ALPHA)**2*C1+BETA-1.D+0
FA=-2.D+0*BETA**2*(1.D+0-ALPHA)*C1
FB=2.D+0*BETA*(1.D+0-ALPHA)**2*C1+1.D+0
G=ALPHA**2*BETA*C2+ALPHA-1.D+0
GA=2.D+0*ALPHA*BETA*C2+1.D+0
GB=ALPHA**2*C2
DENOM=FA*GB-FB*GA
DA=(-F*GB+G*FB)/DENOM
DB=(-G*FA+F*GA)/DENOM
ALPHA=ALPHA+DA
BETA=BETA+DB
IF((DABS(DA).GT.1.D-05).OR.(DABS(DB).GT.1.D-5)) GO TO 132
ALPH=RCON(K,9,J)/(RCON(K,9,J)+RCON(K,3,J))
BET=(RCON(K,9,J)+RCON(K,3,J))/(2.D+0*RCON(K,1,J)+
* RCON(K,9,J)+RCON(K,3,J))
IF(IQ6.EQ.1) GO TO 1888
BETN(K,J)=(BET**2/(1.D+0-BET))*((1.D+0-BETA)/(BETA**2))
* ((1.D+0-ALPH)**2/(1.D+0-ALPHA)**2)
1888 ALPN(K,J)=BET*(ALPH**2/(1.D+0-ALPH))
* ((1.D+0-ALPHA)/(BETA*ALPHA**2))
122 CONTINUE
DO 142 K=3,NXCON
DO 142 J=1,NPTS(K)
IF(J.EQ.NPTS(K)) THEN
ALPO(K,J)=0.D+00
BETO(K,J)=0.D+00
GO TO 142
ENDIF
Q1=0.D+0
Q2=0.D+0
Q3=0.D+0
DO 152 III=1,LI(4)
152 Q1=Q1+GIL(III,4)*EXP(-EPSIIL(III,4)/TEMP(K,J))
DO 153 III=1,LI(10)
153 Q2=Q2+GIL(III,10)*DEXP(-EPSIIL(III,10)/TEMP(K,J))
DO 160 III=1,LI(2)
160 Q3=Q3+GIL(III,2)*DEXP(-EPSIIL(III,2)/TEMP(K,J))
RHO1=RCON(K,4,J)*1.600D+01/6.023D+23

```

```

RHO2=RCON(K,10,J)*1.600D+01/6.023D+23
RHO3=RCON(K,2,J)*3.200D+01/6.023D+23
RHO4=RHO1+RHO2+RHO3
QVIB=1.D+0/(1.D+0-DEXP(-2.27D+03/TEMP(K,J)))
QROT=TEMP(K,J)/4.2D+0
C1=RHO4*1.7725D+01*QVIB*QROT*Q3*DEXP(5.950D+04/TEMP(K,J))
C1=C1/(Q1**2*TEMP(K,J)**1.5)
C2=RHO4*7.801D+06*Q1*DEXP(1.58D+05/TEMP(K,J))
C2=C2/(Q2*TEMP(K,J)**1.5)
ALPHA=0.5D+0
BETA=0.5D+0
162 F=BETA**2*(1.D+0-ALPHA)**2*C1+BETA-1.D+0
FA=-2.D+0*BETA**2*(1.D+0-ALPHA)*C1
FB=2.D+0*BETA*(1.D+0-ALPHA)**2*C1+1.D+0
G=ALPHA**2*BETA*C2+ALPHA-1.D+0
GA=2.D+0*ALPHA*BETA*C2+1.D+0
GB=ALPHA**2*C2
DENOM=FA*GB-FB*GA
DA=(-F*GB+G*FB)/DENOM
DB=(-G*FA+F*GA)/DENOM
ALPHA=ALPHA+DA
BETA=BETA+DB
IF((DABS(DA).GT.1.D-05).OR.(DABS(DB).GT.1.D-5)) GO TO 162
ALPH=RCON(K,10,J)/(RCON(K,10,J)+RCON(K,4,J))
BET=(RCON(K,10,J)+RCON(K,4,J))/(2.D+0*RCON(K,2,J)+
* RCON(K,10,J)+RCON(K,4,J))
IF(IQ6.EQ.1) GO TO 1999
BETO(K,J)=(BET**2/(1.D+0-BET))*((1.D+0-BETA)/(BETA**2))
* ((1.D+0-ALPH)**2/(1.D+0-ALPHA)**2)
1999 ALPO(K,J)=BET*(ALPH**2/(1.D+0-ALPH))
* ((1.D+0-ALPHA)/(BETA*ALPHA**2))
142 CONTINUE
C BETA=BETA(Tt)
IF(IQ6.EQ.1) THEN
DO 1122 K=3,NXCON
DO 1122 J=1,NPTS(K)
IF(J.EQ.NPTS(K)) THEN
BETN(K,J)=0.D+00
GO TO 1122
ENDIF
Q1=0.D+0
Q2=0.D+0
Q3=0.D+0
DO 1112 III=1,LI(3)
1112 Q1=Q1+GIL(III,3)*DEXP(-EPSIIL(III,3)/TEMPO(K,J))
DO 1113 III=1,LI(9)
1113 Q2=Q2+GIL(III,9)*DEXP(-EPSIIL(III,9)/TEMPO(K,J))
DO 1130 III=1,LI(1)
1130 Q3=Q3+GIL(III,1)*DEXP(-EPSIIL(III,1)/TEMPO(K,J))
RHO1=RCON(K,3,J)*1.401D+01/6.023D+23
RHO2=RCON(K,9,J)*1.401D+01/6.023D+23
RHO3=RCON(K,1,J)*2.802D+01/6.023D+23
RHO4=RHO1+RHO2+RHO3
QVIB=1.D+0/(1.D+0-DEXP(-3.39D+03/TEMPO(K,J)))
QROT=TEMPO(K,J)/5.8D+0
C1=RHO4*2.4701D+01*QVIB*QROT*Q3*DEXP(1.135D+05/TEMPO(K,J))
C1=C1/(Q1**2*TEMPO(K,J)**1.5)
C2=RHO4*8.909D+06*Q1*DEXP(1.69D+05/TEMPO(K,J))
C2=C2/(Q2*TEMPO(K,J)**1.5)
ALPHA=0.5D+0

```

```

      BETA=0.5D+0
1132  F=BETA**2*(1.D+0-ALPHA)**2*C1+BETA-1.D+0
      FA=-2.D+0*BETA**2*(1.D+0-ALPHA)*C1
      FB=2.D+0*BETA*(1.D+0-ALPHA)**2*C1+1.D+0
      G=ALPHA**2*BETA*C2+ALPHA-1.D+0
      GA=2.D+0*ALPHA*BETA*C2+1.D+0
      GB=ALPHA**2*C2
      DENOM=FA*GB-FB*GA
      DA=(-F*GB+G*FB)/DENOM
      DB=(-G*FA+F*GA)/DENOM
      ALPHA=ALPHA+DA
      BETA=BETA+DB
      IF((DABS(DA).GT.1.D-05).OR.(DABS(DB).GT.1.D-5)) GO TO 1132
      BET=(RCON(K,9,J)+RCON(K,3,J))/(2.D+0*RCON(K,1,J)+
*      RCON(K,9,J)+RCON(K,3,J))
      ALPH=RCON(K,9,J)/(RCON(K,9,J)+RCON(K,3,J))
      BETN(K,J)=(BET**2/(1.D+0-BET))*((1.D+0-BETA)/(BETA**2))
*      *((1.D+0-ALPH)**2/(1.D+0-ALPHA)**2)
1122  CONTINUE
      DO 1142 K=3,NXCON
      DO 1142 J=1,NPTS(K)
      IF(J.EQ.NPTS(K)) THEN
        BETO(K,J)=0.D+00
        GO TO 1142
      ENDIF
      Q1=0.D+0
      Q2=0.D+0
      Q3=0.D+0
      DO 1152 III=1,LI(4)
1152  Q1=Q1+GIL(III,4)*EXP(-EPSIIL(III,4)/TEMPO(K,J))
      DO 1153 III=1,LI(10)
1153  Q2=Q2+GIL(III,10)*DEXP(-EPSIIL(III,10)/TEMPO(K,J))
      DO 1160 III=1,LI(2)
1160  Q3=Q3+GIL(III,2)*DEXP(-EPSIIL(III,2)/TEMPO(K,J))
      RHO1=RCON(K,4,J)*1.600D+01/6.023D+23
      RHO2=RCON(K,10,J)*1.600D+01/6.023D+23
      RHO3=RCON(K,2,J)*3.200D+01/6.023D+23
      RHO4=RHO1+RHO2+RHO3
      QVIB=1.D+0/(1.D+0-DEXP(-2.27D+03/TEMPO(K,J)))
      QROT=TEMPO(K,J)/4.2D+0
      C1=RHO4*1.7725D+01*QVIB*QROT*Q3*DEXP(5.950D+04/TEMPO(K,J))
      C1=C1/(Q1**2*TEMPO(K,J)**1.5)
      C2=RHO4*7.801D+06*Q1*DEXP(1.58D+05/TEMPO(K,J))
      C2=C2/(Q2*TEMPO(K,J)**1.5)
      ALPHA=0.5D+0
      BETA=0.5D+0
1162  F=BETA**2*(1.D+0-ALPHA)**2*C1+BETA-1.D+0
      FA=-2.D+0*BETA**2*(1.D+0-ALPHA)*C1
      FB=2.D+0*BETA*(1.D+0-ALPHA)**2*C1+1.D+0
      G=ALPHA**2*BETA*C2+ALPHA-1.D+0
      GA=2.D+0*ALPHA*BETA*C2+1.D+0
      GB=ALPHA**2*C2
      DENOM=FA*GB-FB*GA
      DA=(-F*GB+G*FB)/DENOM
      DB=(-G*FA+F*GA)/DENOM
      ALPHA=ALPHA+DA
      BETA=BETA+DB
      IF((DABS(DA).GT.1.D-05).OR.(DABS(DB).GT.1.D-5)) GO TO 1162
      BET=(RCON(K,10,J)+RCON(K,4,J))/(2.D+0*RCON(K,2,J)+
*      RCON(K,10,J)+RCON(K,4,J))

```

```

      ALPH=RCON(K,10,J)/(RCON(K,10,J)+RCON(K,4,J))
      BETO(K,J)=(BET**2/(1.D+0-BET))*(1.D+0-BETA)/(BETA**2))
      *
      *((1.D+0-ALPH)**2/(1.D+0-ALPHA)**2)
1142  CONTINUE
      ENDIF
      ENDIF
C      ABSORPTION COEFFICIENTS
      DO 20 K=3,NXCON
      DO 20 J=1,NPTS(K)
      TT=TEMP(K,J)/168800.0
      IF(IQ6.EQ.1) THEN
        TM=TEMPO(K,J)/168800.0
      ELSE
        TM=TT
      ENDIF
C      IF((IQ6.EQ.1).AND.(TEMPO(K,J).LE.8.D+03)) GO TO 20
C      IF(TEMP(K,J).LE.8.D+03) GO TO 20
      IF(IQ8.EQ.0) THEN
        O=BETO(K,J)
        N=BETN(K,J)
      ELSE
        O=1.D+0
        N=1.D+0
      ENDIF
      KAPPA(K,4,J)=5.D-19*O*RCON(K,2,J)+5.D-20*N*RCON(K,1,J)
      KAP(K,4,J)=1.7D-17*RCON(K,3,J)*DEXP(-.246/TT)
      KA(K,4,J)=0.D+0
      KAPPA(K,3,J)=2.D-18*(N*RCON(K,1,J)+O*RCON(K,2,J))+KAPPA(K,4,J)
      KAP(K,3,J)=2.1D-17*RCON(K,3,J)*DEXP(-.165/TT)+KAP(K,4,J)
      KA(K,3,J)=0.D+0
      KAPPA(K,2,J)=5.1D-18*(N*RCON(K,1,J)+O*RCON(K,2,J))+KAPPA(K,3,J)
      KAP(K,2,J)=KAP(K,3,J)
      KA(K,2,J)=5.1D-18*RCON(K,4,J)
      KAPPA(K,1,J)=2.D-17*O*RCON(K,2,J)+4.D-16*N*RCON(K,1,J)
C      +KAPPA(K,2,J)
      KAP(K,1,J)=1.1D-17*RCON(K,3,J)+KAP(K,2,J)
      KA(K,1,J)=KA(K,2,J)
      IF(IQ5.EQ.0) THEN
        N=ALPN(K,J)
        O=ALPO(K,J)
      IF(IQ8.EQ.0) THEN
        NN=BETN(K,J)
        OO=BETO(K,J)
      ELSE
        NN=1.D+0
        OO=1.D+0
      ENDIF
      KAPPA(K,5,J)=7.7D-17*(NN*RCON(K,1,J)+OO*RCON(K,2,J))
C      *DEXP(-.49/TT)+
C      2.6D-17*(RCON(K,3,J)*N+RCON(K,4,J)*O)*DEXP(-.723/TT)
      KAPPA(K,6,J)=2.D-18*RCON(K,2,J)*OO+6.0D-18*(RCON(K,3,J)*N+
C      RCON(K,4,J)*O)*DEXP(-.379/TT)+KAPPA(K,5,J)
      IF(RCON(K,7,J).LE.1.D+00) THEN
        KAPPA(K,7,J)=KAPPA(K,6,J)
        GO TO 909
      ENDIF
      KAPPA(K,7,J)=1.2D03*((RCON(K,3,J)*N+RCON(K,4,J)*O)/RCON(K,7,J))*
C      DEXP(-.489/TT)+KAPPA(K,6,J)
909  KAPPA(K,8,J)=3.2D-17*(RCON(K,3,J)*N+RCON(K,4,J)*O)*DEXP(-.631/TT)
C      +KAPPA(K,5,J)

```



```

ELSE
  KAPPA(K,5,J)=7.7D-17*(RCON(K,1,J)+RCON(K,2,J))*DEXP(-.49/TT)+
C      2.6D-17*(RCON(K,3,J)+RCON(K,4,J))*DEXP(-.723/TT)
  KAPPA(K,6,J)=2.D-18*RCON(K,2,J)+6.0D-18*(RCON(K,3,J)+RCON(K,4,J))
C      *DEXP(-.379/TT)+KAPPA(K,5,J)
  IF(RCON(K,7,J).LE.1.D+00) THEN
    KAPPA(K,7,J)=KAPPA(K,6,J)
    GO TO 910
  ENDIF
  KAPPA(K,7,J)=1.2D03*((RCON(K,3,J)+RCON(K,4,J))/RCON(K,7,J))*
C      DEXP(-.489/TT)+KAPPA(K,6,J)
910 KAPPA(K,8,J)=3.2D-17*(RCON(K,3,J)+RCON(K,4,J))*DEXP(-.631/TT)
C      +KAPPA(K,5,J)
  ENDIF
20 CONTINUE

```

PLANCK FUNCTION

```

C
DO 30 K=3,NXCON
DO 30 J=1,NPTS(K)
TT=TEMP(K,J)/168800.0
C
IF(TEMP(K,J).LE.8.D+03) GO TO 30
BLACK=(TT*168800.)*4*5.6696D-12/3.1415927
FF=15.D+00/(3.1415927*4)
X1=1./TT
B1=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C  DEXP(-2.*X1)*0.5*(.75+1.5*X1+1.5*X1**2+X1**3))
X1=0.935/TT
B2=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C  DEXP(-2.*X1)*0.5*(.75+1.5*X1+1.5*X1**2+X1**3))
X1=0.835/TT
B3=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C  DEXP(-2.*X1)*0.5*(.75+1.5*X1+1.5*X1**2+X1**3))
X1=0.754/TT
B4=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C  DEXP(-2.*X1)*0.5*(.75+1.5*X1+1.5*X1**2+X1**3))
X1=0.473/TT
B6=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C  DEXP(-2.*X1)*0.5*(.75+1.5*X1+1.5*X1**2+X1**3))
X1=0.213/TT
B8=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C  DEXP(-2.*X1)*0.5*(.75+1.5*X1+1.5*X1**2+X1**3))
B8=1-B8
B2=B2-B1
B3=B3-B2
B4=B4-B3
B6=B6-B4
B5=1-B8-B6-B4-B3-B2-B1
B(K,1,J)=BLACK*B1
B(K,2,J)=BLACK*B2
B(K,3,J)=BLACK*B3
B(K,4,J)=BLACK*B4
B(K,6,J)=BLACK*B6
B(K,7,J)=BLACK*2.4D-21*RCON(K,7,J)*B6*DEXP(.162/TT)
B(K,8,J)=BLACK*B8
B(K,5,J)=BLACK*B5
30 CONTINUE

```

TAU'S

```

C
DO 100 K=3,NXCON
DO 100 I=1,8
IF(I.GT.4) THEN
  OPTL(K,I,1)=KAPPA(K,I,1)*DABS(Y(K,1)-YBDY(K))

```



```

150 CONTINUE
    WRITE(*,103) I,SUM
    WRITE(8,103) I,SUM
103 FORMAT(/,' FOR BAND ',I2,' QR = ',D11.4,' WATTS/SQ.CM.')
```

WQR=WQR+SUM

```

180 CONTINUE
    WRITE(*,101) WQR
    WRITE(8,101) WQR
101 FORMAT(/' TOTAL QR = ',D11.4,' WATTS/SQ.CM.')
```

```

200 CONTINUE
    RETURN
    END
```

C

C

```

SUBROUTINE FEI(E1I,E2I,E3I,Z)
IMPLICIT REAL*8(A-H,O-Z)
COMMON /RAD1/ RRHO(30,30),Y(30,30),YBDY(30)
COMMON /RAD3/ B(30,8,30)
COMMON /RAD5/ OPTL(30,8,30)
REAL OPTL,XPST,Y,YBDY,TEMPO,TEMP,Z
REAL ALPN,ALPO,B,BETN,BETO
CALL EXPI(Z,E1I,AUX)
E2I=DEXP(-Z)-Z*E1I
E3I=(DEXP(-Z)-Z*E2I)/2.0
E2I=DEXP(-Z*DSQRT(3.D+00))
RETURN
END
```

C

C

C

```

SUBROUTINE EXPI(X,RES,AUX)
IMPLICIT REAL*8(A-H,O-Z)
COMMON /RAD1/ RRHO(30,30),Y(30,30),YBDY(30)
COMMON /RAD3/ B(30,8,30)
COMMON /RAD5/ OPTL(30,8,30)
REAL OPTL,XPST,Y,YBDY,TEMPO,TEMP,Z
REAL ALPN,ALPO,B,BETN,BETO
IF(X-1.) 2,1,1
1  YY=1./X
   AUX=1.-YY*(((YY+3.377358)*YY+2.052156)*YY+2.709479D-01)/(((YY*
C1.072553+5.716943)*YY+6.945239)*YY+2.593888)*YY+2.709496D-01)
   RES=AUX*YY*DEXP(-X)
   RETURN
2  IF(X+3.) 6,6,3
3  AUX=((((((7.122452D-7*X-1.766345D-6)*X+2.928433D-5)*X-2.335379D-4
C)*X+1.664156D-3)*X-1.041576D-2)*X+5.555682D-2)*X-2.500001D-1)*X
C+9.999999D-1
   RES=-1.D+30
   IF(X) 4,5,4
4  RES=X*AUX-DLOG(ABS(X))-5.772157D-1
5  RETURN
6  IF(X+9.) 8,8,7
7  AUX=1.-((((5.176245D-2*X+3.061037)*X+3.243655D+1)*X+2.244234D+2)*X
C+2.486697D+2)/((((X+3.995161)*X+3.893944D+1)*X+2.263818D+1)*X
C+1.807837D+2)
   GO TO 9
8  YY=9./X
   AUX=1.-YY*(((YY+7.659824D-1)*YY-7.271015D-1)*YY-1.080693)/(((YY
C*2.518750+1.122927D+1)*YY+5.921405)*YY-8.666702)*YY-9.724216)
```

9 RES-AUX*DEXP(-X)/X
RETURN
END

STATEMENT LISTING OF PROGRAM AFEM

PROGRAM AFEM - (IBM-3090, "WYLBUR" VERSION)

AN INVERSE HYPERSONIC FLOW SOLUTION FOR AN AFE/AOTV BODY IN
CHEMICAL NONEQUILIBRIUM USING COUPLED VIBRATION-DISSOCIATION MODELS,
SHOCK JUMP APPROXIMATIONS, ELECTRON TEMPERATURE MODELING, AND
UNCOUPLED RADIANT HEAT TRANSFER USING A METHOD DERIVED BY S.MASLEN

BASED ON AN ORIGINAL PROGRAM BY B.L. WEIGEL FOR WILLIAM L. GROSE
VIRGINIA POLYTECHNIC INSTITUTE

MAJOR PROGRAM REVISIONS BY G.BOBBSKILL, R.GREENDYKE, AND L.CARLSON
TEXAS A&M UNIVERSITY

CALITH : INTEGRATION ROUTINE BY CHARLES E. TREANOR
THE NUMERICAL INTEGRATION ALGORITHM USED IS FOUND IN A METHOD
FOR THE NUMERICAL INTEGRATION OF COUPLED FIRST ORDER
DIFFERENTIAL EQUATIONS WITH GREATLY DIFFERENT TIME CONSTANTS

FOFE - EVALUATE E BY NEWTON ITERATION METHOD
FOFTS1 - EVALUATE TS BY NEWTON ITERATION METHOD, CHEMISTRY FROZEN
FOFT23 - FOFTS1 WITH N2 FROZEN, O2 DISSOCIATING OR BOTH N2 & O2 DISS.
FTLUP - INTERPOLATION ROUTINE

DATA FILES

UNIT 5 IS USED TO READ DATA FROM FILE SCREEN
UNIT 7 IS USED TO READ DATA FROM INPUT DATA FILE
UNIT 8 IS USED FOR THE OUTPUT DATA FILE
UNIT 9 IS USED TO STORE VIBRATIONAL ENERGY DATA AT THE SHOCK (FORMATTED)
UNIT 10 IS USED TO STORE ALL STAGNATION QUANTITIES
UNIT 11 IS USED TO STORE ALL EVIS DATA FOR COMPUTATIONAL PURPOSES
UNIT 12 IS USED TO STORE ALL PRESSURE DATA
UNIT 13 IS USED TO STORE ALL DPDX DATA
UNIT 14 IS USED TO STORE QUANTITIES FOR PHYSICAL SPACE CALCULATIONS
UNIT 15 IS USED TO STORE ALL WARNINGS DURING PROGRAM EXECUTION

INPUT-NAMELIST

DELX = INCREMENT ALONG SHOCK , cm
ZSTERM = LENGTH OF SYMMETRY AXIS , Z , cm
IMAX = MAX. NO. OF I-S SPECIES , LESS THAN OR EQUAL TO 25
JMAX = MAX. NO. OF J-S REACTIONS , LESS THAN OR EQUAL TO 50
MJ = CODE INDICATING WHICH SPECIES , I , TO USE TO CALCULATE
COUPLING FACTOR, PHI SUB J , FOR REACTION J
M = 1 FOR VIBRATIONAL NON-EQUILIBRIUM
= 0 FOR VIBRATIONAL EQUILIBRIUM
R = UNIVERSAL GAS CONSTANT , erg/(mole-K)
GAMMA = RATIO OF SPECIFIC HEATS
CIINF = FREE STREAM MASS FRACTION FOR EACH SPECIES
PINF = FREE STREAM PRESSURE , dynes/cm**2
TINF = FREE STREAM TEMPERATURE , K
VINP = FREE STREAM VELOCITY , cm/sec
MUI = MOLECULAR WT. FOR EACH SPECIES , gm/mole
THETA1 = CHARACTERISTIC VIBRATIONAL TEMPERATURE , K
DGEN1 = FUDGE FACTOR TO PERMIT APPROXIMATING POLYATOMIC MOLECULE
BY A DIATOMIC MOLECULE
FI = 0 FOR MONATOMIC SPECIES
= 1 FOR ALL OTHERS
DELHI = HEAT OF FORMATION , ergs/mole
DELI = DISSOCIATION ENERGY OF SPECIES , K
EVI = VIBRATIONAL ENERGY OF SPECIES , ergs/g
BI = $3/2 \ln(2\pi M_i k/h^2) + \ln(k) + FI(I) \ln(THETA1(Rotational))$
+ $\ln(GIL(I,1))$ FOR EACH SPECIE I
LI = NUMBER OF ELECTRONIC LEVELS FOR EACH SPECIES (LI.LE.20)
GIL = DEGENERACY OF L-TH ELECTRONIC LEVEL FOR I-TH SPECIES
EPSIIL = L-TH ELECTRONIC ENERGY LEVEL FOR I-TH SPECIES , K
AJ = FREQUENCY FACTOR IN ARRENIUS TYPE RATE EQN.
BJ = TEMPERATURE EXPONENT IN ARRENIUS TYPE EQN.
EJ = ACTIVATION ENERGY IN ARRENIUS TYPE EQN.
DIRECT = DIRECTION OF THE REACTION (FORW. = 1.0 , BACK. = 2.0)
AIJ = FACTOR TO ALLOW USE OF GENERAL SPECIES IN REACTION EQNS.
i.e. (N2 + M = 2N + M), AIJ = 1.0 OR (N2 + O2 = 2NO), AIJ = 0.0
NUIJ = STOICHIOMETRIC COEFFICIENTS OF I-TH REACTANT IN J-TH REACTION
NUIJ = STOICHIOMETRIC COEFFICIENT FOR I-TH PRODUCT IN J-TH REACTION
ALPIK = FACTORS IN EQN. FOR VIBRATIONAL RELAXATION TIME
BETAIK = FACTORS IN EQN. FOR VIBRATIONAL RELAXATION TIME

```

C      SIGIK  = FACTORS IN EQN. FOR VIBRATIONAL RELAXATION TIME
C      NIP    = NUMBER OF VIBRATIONAL LEVELS FOR ANHARMONIC OSCILLATOR , K
C      UP     = CHARACTERISTIC PROBABILITY TEMPERATURE , K
C      WE     = CONST. USED TO COMPUTE ANHARMONIC OSCILLATOR , K
C      WEXE   = CONST. USED TO COMPUTE ANHARMONIC OSCILLATOR , K
C      WEYE   = CONST. USED TO COMPUTE ANHARMONIC OSCILLATOR , K
C      WEZE   = CONST. USED TO COMPUTE ANHARMONIC OSCILLATOR , K
C      XI     = INITIAL COMPUTING INTERVAL , .0001220703125 UNLESS INPUT , cm
C      ELE1   = (2*IMAX + 1) VALUES USED BY INTEGRATION SCHEME
C              NORMALLY 0.1,0.5, OR .05
C      ELE2   = (2*IMAX + 1) VALUES USED BY INTEGRATION SCHEME
C              NORMALLY .05,.1, OR .01 AND .LT. ELE1
C      XPST   = 99 OR LESS Xs AT WHICH PHYSICAL SPACE CALCULATIONS ARE
C              DESIRED. THEY MUST BE MULTIPLES OF DELX IN ORDER TO HAVE
C              RS,COST,ZS, AND SINT VALUES AND LAST MUST BE .GT. X AT
C              ZSTERM. THEREFORE, XPST(NXPST) SET = X AT ZSTERM + 100.0
C              IN PROGRAM. XPST(1) MAY NOT BE 0.0 . THEREFORE, SET
C              XPST(1) = DELX IN PROGRAM
C      NXPST  = NUMBER OF X-S AT WHICH PHYSICAL SPACE CALCULATIONS ARE
C              DESIRED
C      CIMAX  = MAX. CJ OR COMPUTING INTERVAL 0.0625 UNLESS INPUT OTHERWISE
C      HCHCKT = CONTROL ON SIZE OF COMPUTING INTERVAL IN CHECK
C              IF(ABS(HPREV - H)/H.GT.HCHECK) REDUCE INTERVAL
C      TCHCKT = CONTROL ON SIZE OF COMPUTING INTERVAL IN CHECK
C              IF(ABS(TPREV - T)/T.GT.TCHECK) REDUCE INTERVAL
C      PHMAX  = CONTROL ON COMPUTING INTERVAL .LE. 65.0
C      IPF    = OUTPUT PRINT FREQUENCY

C      STOPS
C      -----
C      STOP 1   INCORRECT INPUT
C      STOP 2   IN MAIN WHEN NXPST IS .LE. 2
C      STOP 13  IN SHOCKG
C      STOP 30  IN CHECK WHEN COMPUTING INTERVAL .LT. 1.OE-15
C      STOP 66  IN BASIC WHEN NO CONVERGENCE ON E ITERATION (FOFE)
C      STOP 301 IN MAIN FOR ERROR IN XPST ARRAY OR IZTERM .GT. 500
C      STOP 321 IN MAIN WHEN NO CONVERGENCE ON TS ITERATION (FOFTS)
C      STOP 663 IN MAIN WHEN X .NE. VARI(IPSI)
C      STOP 665 IN MAIN AFTER AN INTEGRATION ATTEMPT
C      STOP 670 IN MAIN WHEN A CI NEGATIVE

C      IMPLICIT REAL*8(A-H,O-Z)
C      COMMON /A1/ TE1,TE2,M,IX,IMAX,IPSI,MODEL
C      COMMON /A2/ P(500),DPDX(500),VARI(500)
C              FOLLOWING 7 VARIABLES DIMENSIONED BY IMAX
C      COMMON /A3/ EVIINF(25),THETAI(25),MUI(25),FI(25),DELHI(25),
C      *      CIINF(25),LI(25)
C      COMMON /A4/ R,MUINF,DELTA,LAMSQ,OMEGSQ,OMEGA,MU,T,U,TINF
C              FOLLOWING 2 VARIABLES DIMENSIONED BY (LMAX IN LI,IMAX)
C      COMMON /A5/ EPSIIL(20,25),GIL(20,25),CID(200,50)
C      COMMON /A6/ VAR(52),CUVAR(52),DER(51)
C              FOLLOWING 4 VARIABLES DIMENSIONED BY JMAX
C      COMMON /A7/ MJ(50),EJ(50),AJ(50),BJ(50),DIRECT(50)
C              FOLLOWING 3 VARIABLES DIMENSIONED BY IMAX
C      COMMON /A8/ TVI(25),DGENI(25),BI(25),NI(25)
C              EVIS(IMAX)
C      COMMON /A9/ EVIS(25),XPST(100)
C              FOLLOWING 3 VARIABLES DIMENSIONED BY (IMAX+1,JMAX) OR (IMAX,JMAX)
C      COMMON /A10/ NUIJ(26,50),AIJ(25,50),NUPIJ(25,50)
C              FOLLOWING 3 VARIABLES DIMENSIONED BY (IMAX,JMAX)
C      COMMON /A11/ SIGIK(25,25),ALPIK(25,25),BETAIK(25,25)
C      COMMON /A12/ SINTM(1500),COSTM(1500),RSM(1500),RCM(1500),X1(1500)
C      *      ,ZSM(1500)
C      COMMON /A13/ SP,TS,DELX,ZSTERM,IZTERM,NSR,MW
C      COMMON /A14/ EINF,PINF,RHOINF,VINF,E,JMAX,KEYINT,RHO,HSTAG
C      COMMON /A15/ PFTL,KITR1,NIP(25),UP(25)
C      COMMON /A16/ WE(25),WEXE(25),WEYE(25),WEZE(25),IUNEG
C      COMMON /A17/ ELB,SPEC,CJ,TPREV,HPREV,HCHECK,TCHECK
C      COMMON /A18/ ITNEG,IEXP
C      COMMON /A19/ ELE1(51),ELE2(51),NERR
C      COMMON /RAD/ RCON(100,10,40),RRHO(100,40),TEMP(100,40)
C      *      ,TSTAG,Y(100,40),NXCON,NPTS(100)
C      COMMON /RADC/ YBDY(100),B(100,8,40),OPTL(100,8,40)
C      COMMON /RADO/ TEMPO(100,40),ALPN(100,40),ALPO(100,40)
C      COMMON /RADDL/ BETN(100,40),BETO(100,40)
C      COMMON /PLOT/ XSTR1(2000),ZSTR1(2000),RSTR1(2000)
C              DELI(IMAX)

C      DIMENSION DELI(25),PREC(25)
C      DIMENSION RUT(200),PSISTG(200)
C      DIMENSION CIG(25,11),EIG(25,11)

```



```

DO 45 J = 1, JMAX, 2
45 READ(7,*) MJ(J), AJ(J), BJ(J), EJ(J), DIRECT(J)
DO 48 J = 2, JMAX, 2
48 READ(7,*) MJ(J), DIRECT(J)
IMAX21 = 2*IMAX + 1
DO 51 I = 1, IMAX21
51 READ(7,*) ELE1(I), ELE2(I)
DO 54 I = 1, IMAX
54 READ(7,*) (AIJ(I,J), J = 1, JMAX)
DO 57 J = 1, JMAX
57 READ(7,*) (NUIJ(I,J), I = 1, IMAX+1)
DO 60 J = 1, JMAX
60 READ(7,*) (NUIJ(I,J), I = 1, IMAX)
DO 63 I = 1, IMAX
63 READ(7,*) (ALPIK(J,I), J = 1, IMAX)
DO 66 I = 1, IMAX
66 READ(7,*) (BETAIK(J,I), J = 1, IMAX)
DO 69 I = 1, IMAX
69 READ(7,*) (SIGIK(J,I), J = 1, IMAX)
DO 72 I = 1, IMAX
LII = LI(I)
72 READ(7,*) (GIL(L,I), L = 1, LII)
DO 75 I = 1, IMAX
LII = LI(I)
75 READ(7,*) (EPSIIL(L,I), L = 1, LII)
IF (IANS.EQ.1) THEN
  NXPST = 4
  XPST(1) = (IS3 - NSR)*DELX
  XPST(2) = (IS4 - NSR)*DELX
  XPST(3) = (IS5 - NSR)*DELX
  XPST(4) = (IS6 - NSR)*DELX
  GO TO 78
ENDIF
DO 76 I = 1, NXPST
NPTS(I) = I
76 XPST(I) = DELX*I
78 DO 80 I = 1, IMAX
80 READ(7,*) NIP(I), UP(I), WE(I), WEXE(I), WEYE(I), WEZE(I)
IMAXP2 = IMAX + 2

C                                SUM OF CIINF SHOULD BE 1.0
SUM = 0.0
DO 81 I = 1, IMAX
81 SUM = SUM + CIINF(I)
IF (SUM.EQ.1.0) GO TO 85
WRITE(15,82) SUM
82 FORMAT(1X,'SUM OF CIINF(I) SHOULD BE 1: IT IS = ',E11.4,' STOP 1')
STOP

C                                STOP 1
85 HCHECK = HCHCKT
TCHECK = TCHCKT

C                                READ IN AND WRITE OUT SPECIES AND REACTIONS
DO 88 I = 1, IMAX
88 READ(7,90) SPECIE(I)
90 FORMAT(A30)
WRITE(8,95)
95 FORMAT(//)
WRITE(8,100)
100 FORMAT(14X,'SPECIES'/)
DO 110 I = 1, IMAX
105 FORMAT(1X,I5,11X,A30)
110 WRITE(8,105) I,SPECIE(I)
WRITE(8,95)
DO 112 I = 1, JMAX
112 READ(7,90) REACT(I)
WRITE(8,115)
115 FORMAT(14X,'REACTIONS'/)
DO 120 I = 1, JMAX
120 WRITE(8,125) I,REACT(I)
125 FORMAT(1X,I5,8X,A30)
WRITE(8,95)
REWIND 7

C                                STORE PRELIMINARY INFO. FOR DISSPLA
WRITE(69,*) IMAX,JMAX,MODEL,VINF,PINF,TINF

C                                PRINT INPUT
WRITE(8,135) HCHCKT,TCHCKT,PHMAX
135 FORMAT(1X,'HCHCKT = ',E11.4,4X,'TCHCKT = ',E11.4,4X,'PHMAX = ',
*      E11.4/)
WRITE(8,140) XI,CIMAX,NXPST
140 FORMAT(1X,'XI = ',E11.4,8X,'CIMAX = ',E11.4,5X,'NXPST = ',I3//)
DO 141 I = 1, IMAX21

```

```

141 WRITE(8,142) I,ELE1(I),I,ELE2(I)
142 FORMAT(1X,'ELE1(',I2,') = ',E11.4,5X,'ELE2(',I2,') = ',E11.4)
    WRITE(8,95)
    DO 143 I = 1,NXPST
143 WRITE(8,144) I,XPST(I)
144 FORMAT(1X,'XPST(',I3,') = ',E11.4)
    WRITE(8,95)
    WRITE(8,145) IMAX,JMAX,M
145 FORMAT(1X,'IMAX = ',I2,11X,'JMAX = ',I2,13X,'M = ',I2/)
    WRITE(8,150) DELX,ZSTERM,IPF,R,PINF,TINF,VINF,GAMMA
150 FORMAT(1X,'DELX = ',E11.4,2X,'ZSTERM = ',E11.4,2X,
* 'PRINT FREQ. = ',I3,3X,'R = ',E11.4,/,1X,'PINF = ',E11.4,
* 2X,'TINF = ',E11.4,4X,'VINF = ',E11.4,2X,'GAMMA = ',E9.3//)
    WRITE(8,155)
155 FORMAT(16X,'MUI',8X,'THETAI',7X,'DGENI',10X,'FI',9X,'DELHI'//)
    DO 157 I = 1,IMAX
157 WRITE(8,160) I,MUI(I),THETAI(I),DGENI(I),FI(I),DELHI(I)
160 FORMAT(1X,I2,8X,5(E11.4,2X))
    WRITE(8,95)
    WRITE(8,161)
161 FORMAT(16X,'DELI',8X,'CIINF',9X,'EVI',10X,'LI',11X,'BI'//)
    DO 163 I = 1,IMAX
163 WRITE(8,164) I,DELI(I),CIINF(I),EVI(I),LI(I),BI(I)
164 FORMAT(1X,I2,8X,3(E11.4,2X),3X,I3,7X,E11.4)
    WRITE(8,95)
    WRITE(8,165)
165 FORMAT(8X,'NIP',9X,'UP',10X,'WE',10X,'WEXE',9X,'WEYE',8X,'WEZE'//)
    DO 167 I = 1,IMAX
166 FORMAT(1X,I2,5X,I3,4X,E10.4,2X,E10.4,3X,E10.4,3X,E10.4,2X,E10.4)
167 WRITE(8,166) I,NIP(I),UP(I),WE(I),WEXE(I),WEYE(I),WEZE(I)
    WRITE(8,95)
    WRITE(8,170)
170 FORMAT(12X,'MJ',9X,'AJ',14X,'BJ',15X,'EJ',10X,'DIRECTION',/,
* 69X,'F=1,B=2'//)
    DO 180 J = 1,JMAX,2
175 FORMAT(1X,I2,2X,I8,3(2X,E15.6),5X,F5.2)
180 WRITE(8,175) J,MJ(J),AJ(J),BJ(J),EJ(J),DIRECT(J)
    WRITE(8,95)
    WRITE(8,181)
181 FORMAT(12X,'MJ',6X,'DIRECTION',/,21X,'F=1,B=2'//)
    DO 183 J = 2,JMAX,2
182 FORMAT(1X,I2,2X,I8,8X,F5.2)
183 WRITE(8,182) J,MJ(J),DIRECT(J)
    WRITE(8,95)
    DO 187 I = 1,IMAX
    LII = LI(I)
    DO 186 L = 1,LII
    WRITE(8,185) L,I,GIL(L,I),L,I,EPSIIL(L,I)
185 FORMAT(1X,'GIL(',I2,',',I2,') = ',E11.4,4X,
* 'EPSIIL(',I2,',',I2,') = ',E11.4)
186 CONTINUE
187 CONTINUE
    WRITE(8,95)
    DO 200 I = 1,IMAX
    DO 196 J = 1,JMAX
    WRITE(8,195) I,J,AIJ(I,J)
195 FORMAT(1X,'AIJ(',I2,',',I2,') = ',E11.4)
196 CONTINUE
200 CONTINUE
    WRITE(8,95)
    DO 205 I = 1,IMAX+1
    DO 202 J = 1,JMAX
    WRITE(8,201) I,J,NUIJ(I,J),I,J,NUPIJ(I,J)
201 FORMAT(1X,'NUIJ(',I2,',',I2,') = ',I5,9X,'NUPIJ(',I2,',',
* I2,') = ',I5)
202 CONTINUE
205 CONTINUE
    WRITE(8,95)
    DO 225 I = 1,IMAX
    DO 220 J = 1,IMAX
    WRITE(8,210) I,J,SIGIK(I,J),I,J,ALPIK(I,J),I,J,BETAIK(I,J)
210 FORMAT(1X,'SIGIK(',I2,',',I2,') = ',E10.4,1X,
* 'ALPIK(',I2,',',I2,') = ',E10.4,1X,'BETAIK(',I2,',',I2,') = ',
* E10.4)
220 CONTINUE
225 CONTINUE
    WRITE(8,95)
    WRITE(8,230)
230 FORMAT(7X,25('*'), ' END INPUT ',25('*')//)
    WRITE(8,231)

```

```

231 FORMAT(25X,'+++ SHOCK GEOMETRY +++'//)
    NXPSTM = NXPST - 1
C
    CALL SHOCKG (NXPSTM,ITK)
    IZTERM = NO. OF DELTA X INCREMENTS GENERATED IN SHOCKG
C
    WRITE(8,95)
    IF ((NXPST - 1.0).LE.IZTERM) GO TO 237
    WRITE(15,235)
235 FORMAT(1X,'(NXPST-1).GT.IZTERM : IT MUST BE .LE. TO IZTERM',/,
* 1X,'PHYSICAL SPACE CALCULATIONS - SEE DO 700 LOOP : STOP 301'//)
    STOP
237 WRITE(8,95)
    IF (IZTERM.LE.500) GO TO 241
    WRITE(15,240)
240 FORMAT(1X,'IZTERM.GT.500:CHGE DIM OF P,DPDX AND VARI: STOP 301'//)
    STOP
C
    FREESTREAM QUANTITIES
241 SUM = 0.0
    DO 245 I = 1,IMAX
245 SUM = SUM + CIINF(I)/MUI(I)
    MUINF = 1.0/SUM
    RHOINF = MUINF*PINF/(R*TINF)
    AINF = DSQRT(GAMMA*R*TINF/MUINF)
    MINF = VINP/AINF
C
    FOR EACH I , SPECIE
    EINF = 0.0
    DO 275 I = 1,IMAX
    TEM = DEXP(THETAI(I)/TINF)
    EVIINF(I) = (R*THETAI(I))/(MUI(I)*(TEM - 1.0))*FI(I)
C
    FOR EACH L , REACTION LEVEL
    GSUM = 0.0
    GESUM = 0.0
    LII = LI(I)
    IF (LII.LE.20) GO TO 255
    WRITE(15,250) LII
250 FORMAT(1X,'LII = ',I3,2X,'A LEVEL IN LI ARRAY IS GREATER THAN 20',
* 2X,/,1X,'YOU NEED TO CHANGE DIMENSION OF EPSIIL AND GIL')
255 CONTINUE
    DO 265 L = 1,LII
    TEM1 = DEXP(-EPSIIL(L,I)/TINF)
    GSUM = GSUM + GIL(L,I)*TEM1
    GESUM = GESUM + GIL(L,I)*EPSIIL(L,I)*TEM1
265 CONTINUE
    EEIINF = R/MUI(I)*(GESUM/GSUM)
    EIINF = 1.5*R*TINF/MUI(I) + FI(I)*R*TINF/MUI(I) + EVIINF(I)
    * + EEIINF + DELHI(I)/MUI(I)
    EINF = EINF + EIINF*CIINF(I)
275 CONTINUE
    WRITE(8,280) MUINF,RHOINF,AINF,TINF,PINF,VINF,MINF,EIINF,EINF,
    * EEIINF
280 FORMAT(24X,'+++ FREESTREAM QUANTITIES +++',///,6X,'MUINF = ',E11.4
*,3X,'RHOINF = ',E11.4,3X,'AINF = ',E11.4,///,6X,'TINF = ',E11.4,3X
*,3X,'PINF = ',E11.4,3X,'VINF = ',E11.4,///,1X,'MINF = ',E10.4,2X,
*,'EIINF = ',E10.4,2X,'EINF = ',E10.4,2X,'EEIINF = ',E10.4//)
    DO 285 I = 1,IMAX
285 WRITE(8,286) I,EVIINF(I)
286 FORMAT(1X,'EVIINF(',I2,') = ',E11.4)
    WRITE(8,95)
C
    BEGIN QUANTITIES BEHIND SHOCK FOR THE RANGE OF X
    WRITE(8,290)
290 FORMAT(22X,'+++ QUANTITIES BEHIND SHOCK +++',///,5X,'TS',9X,'ES',
*9X,'PS',7X,'RHOS',8X,'US',8X,'PSIS',8X,'HS'//)
    IF (NXPST.GE.3) GO TO 293
    WRITE(15,291)
    STOP
291 FORMAT(1X,'NXPST = 2 - SURELY SOME PHYSICAL SPACE VALUES',/,1X,
* 'ARE DESIRED -- MAKE IT AT LEAST 3 : STOP 2')
C
293 DO 385 IX = 1,IZTERM
    TEM = SINTM(IX)**2
    LAMBDA = RHOINF*VINP*SINTM(IX)
    OMEGA = PINF + RHOINF*VINP**2*TEM
    DELTA = EINF + PINF/RHOINF + (VINP**2*TEM)/2.0
    LAMBSQ = LAMBDA**2
    OMEGSQ = OMEGA**2
    MINFSQ = MINF**2
    TEM = TEM*MINFSQ
    TSG = (TINF*(2.0*GAMMA*TEM - (GAMMA - 1.0))*((GAMMA - 1.0)
* TEM + 2.0))/((GAMMA + 1.0)**2*TEM)
    TOL1 = .001

```

```

TOL2 = .0001
MAXI = 50
C
IF (IQ3.EQ.1) CALL FOFTS1 (MAXI,TOL1,TOL2,ICODE,TSG)
IF (IQ3.GT.1) CALL FOFT23 (MAXI,TOL1,TOL2,ICODE,TSG,IQ1,IQ2,IQ3)
C
IF (ICODE.EQ.1) WRITE(15,330) TSG
330 FORMAT(1X,'**** MAXIMUM ITERATION EXCEEDED IN FOFTS ****',//,
* 1X,'LAST ITERATED VALUE OF TSG WAS ',E15.6,' : STOP 321'//)
IF (ICODE.EQ.2) WRITE(15,335)
335 FORMAT(1X,'***** DERIVATIVE = 0.0 IN FOFTS ***** : STOP 321'//)
IF (ICODE.EQ.1.OR.ICODE.EQ.2) STOP
C
STOP 321
IF (IX.EQ.1) TSTAG = TSG
TS = TSG
ES = 0.0
DO 360 I = 1,IMAX
IF (M.EQ.1) EVIS(I) = EVIINF(I)
IF (M.EQ.1) GO TO 345
TEM = DEXP(THETAI(I)/TS)
EVIS(I) = (R*THETAI(I))/(MUI(I)*(TEM - 1.0))*FI(I)
345 SUMG = 0.0
SUMGE = 0.0
LII = LI(I)
DO 350 L = 1,LII
TEM1 = DEXP(-EPSIIL(L,I)/TS)
SUMG = SUMG + TEM1*GIL(L,I)
350 SUMGE = SUMGE + TEM1*GIL(L,I)*EPSIIL(L,I)
EEIS = (R/MUI(I))*(SUMGE/SUMG)
EIS = (1.5*R*TS)/MUI(I) + (FI(I)*R*TS)/MUI(I) + EVIS(I)
* + EEIS + DELHI(I)/MUI(I)
IF (IQ3.EQ.1) THEN
ES = ES + EIS*CIINF(I)
ELSE
ES = ES + EIS*CID(IX,I)
ENDIF
360 CONTINUE
PS = DSQRT(OMEGSQ - 2.0*LAMSQ*(DELTA - ES))
RHOS = (MUIINF*PS)/(R*TS)
US = VIN*FCOSTM(IX)
PSIS = (RHOINF*VIN*FSM(IX)**2)/2.0
HS = PS/RHOS + ES
WRITE(10,*) IX,TS,ES,PS,RHOS,US,PSIS,HS
WRITE(8,365) TS,ES,PS,RHOS,US,PSIS,HS
365 FORMAT(1X,7(E9.4,2X))
WRITE(9,366) IX
366 FORMAT(1X,'FOR IZTERM = ',I3//)
DO 370 I = 1,IMAX
WRITE(9,367) I,EVIS(I)
367 FORMAT(13X,'EVIS(',I2,') = ',E11.4)
370 CONTINUE
WRITE(11,*) (EVIS(I),I = 1,IMAX)
WRITE(9,620)
385 CONTINUE
C
REWIND 9
REWIND 10
REWIND 11
C
COMPUTE PRESSURE DISTRIBUTION FOR EACH PSI
DO 390 IPSI = 1,IZTERM
DO 388 IX = 1,IZTERM
IF (IX.EQ.IPSI) XX = X1(IX)
READ(10,*) IXT,TS,ES,PS,RHOS,US,PSIS,HS
IF (IXT.LT.IPSI) GO TO 388
IF (IXT.EQ.IPSI) PSISHK = PSIS
IF (IPSI.EQ.IX) THEN
P(IX) = PS
GO TO 388
ENDIF
P(IX) = PS + US/(RCM(IX)*FSM(IX))*(PSISHK - PSIS)
388 CONTINUE
REWIND 10
WRITE(12,*) XX,(P(IX),IX = 1,IZTERM),IPSI,IZTERM
390 CONTINUE
REWIND 12
C
COMPUTE DPDX AND VARI
DO 404 IPSI = 1,IZTERM
READ(12,*) X,(P(I),I=1,IZTERM),IPI,IZTERM
ITHIS = 0
DO 401 IX = IPSI,IZTERM

```



```

KSTAG = 0
ISTAG = 0
XPST(1) = DELX
XPST(NXPST) = VARI(IZTERM) + 100.0
SUMCI = 0.0
WRITE(8,95)
LPS1 = 1
IPSI = 0

C                               BEGIN EACH STREAMLINE COMPUTATION HERE
420 IPSI = IPSI + 1
    IF (IPSI.EQ.IZTERM) GO TO 735
425 KIPF = 0
C                               ISTAG = 2 FOR STREAMLINE DELX
    IF (ISTAG.EQ.1) ISTAG = 2
    READ(10,*) IX,TS,ES,PS,RHOS,US,PSIS,HS
    READ(11,*) (EVIS(I),I = 1,IMAX)
    IF (IPSI.EQ.1) HSTAG = HS
    CJ = XI
    SPEC = 0.0
C                               EVALUATE DERIVATIVES WHEN SPEC = 0.0
C                               II = 0
C                               USE SHOCK VALUES FOR INITIAL COMPUTATION ON EACH STREAMLINE
    KEYINT = 0
    IX = 0
    T = TS
    IF (M.EQ.0) GO TO 450
    DO 445 I = 1,IMAX
445 TVI(I) = TINF
    GO TO 460
450 DO 455 I = 1,IMAX
455 TVI(I) = TS
460 RHO = RHOS
    IF (IQ3.GT.1) THEN
        SUM = 0.0
        DO 462 I = 1,IMAX
462 SUM = SUM + CID(IPSI,I)/MU(I)
        MU = 1.0/SUM
    ENDIF
    IF (IQ3.EQ.1) MU = MUINF
    U = US
    DO 465 I = 1,IMAX
    MM = 2 + I
    K = IMAX + MM
    VAR(MM) = 0.0
    IF (IQ3.EQ.1) VAR(MM) = CIINF(I)
    IF (IQ3.GT.1) VAR(MM) = CID(IPSI,I)
    VAR(K) = 0.0
    IF (M.EQ.0) VAR(K) = EVIS(I)
    IF (M.EQ.1) VAR(K) = EVIINF(I)
    EVI(I) = VAR(K)
465 CONTINUE
C                               COMPUTE S EXPONENT FOR PARK MODEL
    IF (MODEL.EQ.4) THEN
        SP = 3.5*DEXP(-5000.0/TS)
    ENDIF
    E = ES
    H = HS
    IF ((IPSI.EQ.(NSR+1)).AND.(ISTAG.EQ.0)) GO TO 635
    READ(12,*) X,(P(L),L=1,IZTERM),IPSI,IZTERM
    READ(13,*) (DPDX(L),L=1,IZTERM),IPSI,IZTERM
    VAR(1) = X
C                               OMIT PSI = 0.0 STREAMLINE FOR NOW - PICK IT UP LATER
    IF (X.NE.0.0) GO TO 490
    IG = 0
    WRITE(8,725)
    GO TO 730
490 IF (IANS.EQ.1) THEN
    IF ((IPSI.EQ.IS1).OR.(IPSI.EQ.IS2).OR.(IPSI.EQ.IS3)) GO TO 492
    IF ((IPSI.EQ.IS4).OR.(IPSI.EQ.IS5).OR.(IPSI.EQ.IS6)) GO TO 492
    GO TO 730
    ENDIF
492 IF (DABS(X - VARI(IPSI)).LE.1.0E-06) GO TO 500
    WRITE(15,495) IPSI,X,VARI(IPSI)
495 FORMAT(1X,'IPSI = ',I3,2X,'X = ',E10.5,2X,'VARI(IPSI) = ',E10.5,
    *//,1X,'X AND VARI(IPSI) SHOULD BE EQUAL : EXAMINE GENERATION OF X
    * AND VARI IN MAIN : STOP 663'/)
    STOP
C                               STOP 663
C                               INITIALIZE
500 IF (IANS.EQ.1) THEN

```

```

        IF ((IPSI.EQ.IS1).OR.(IPSI.EQ.IS2).OR.(IPSI.EQ.IS3)) GO TO 501
        IF ((IPSI.EQ.IS4).OR.(IPSI.EQ.IS5).OR.(IPSI.EQ.IS6)) GO TO 501
        GO TO 730
    ENDIF
501 ELB = 0.0
    SPEC = 0.0
    IF (VAR(1).GT.XPST(LPS1)) LPS1 = LPS1 + 1
    LPS2 = LPS1
505 IX = IX + 1
C                                     CALITH - MODIFIED RUNGE-KUTTA
    CALL CALITH (N,CIMAX,PHMAX)
C
    IF (NERR.EQ.0) GO TO 575
    IF (NERR.EQ.1) WRITE(15,565) CJ,N
565 FORMAT(1X,'BAD INPUT IN CALITH',/,1X,'CJ = ',E10.5,2X,'N = ',I4
    *,/,1X,'STOP 665'/)
    IF (NERR.EQ.2) WRITE(15,570)
570 FORMAT(1X,'EXAMINE ELE1 AND ELE2 IN CALITH : STOP 665'/)
    IF ((NERR.EQ.1).OR.(NERR.EQ.2)) STOP
C                                     STOP 665
575 IF (IX.EQ.1) THEN
    IF (IQ3.NE.1) THEN
        IF (IQ2.EQ.1) TE = TINF
        IF (IQ2.EQ.2) TE = TS
        TE1 = TE
        TE2 = TE
    ELSE
        TE = TINF
        TE1 = TE
        TE2 = TE
    ENDIF
    ENDIF
C                                     ELECTRON TEMPERATURE MODEL
    IF (IX.EQ.1) GO TO 576
    CORR = 4.23E-06*(T**(-2.88))
    CN = RHO*6.02252E+23
    CNN = CN*VAR(5)/MUI(3)
    CND = CN*VAR(6)/MUI(4)
    TE = DLOG(1. + 10.1/((CND+CNN)*CORR))
    TE = T/(1. + TE*T/85000.)
    TE1 = TE
    CN2 = RHO*6.02252E+23*VAR(3)/MUI(1)
    A = DLOG(1.01D+01) - DLOG(CN2) - DLOG(6.4D+00)
    * - 59.*DLOG(1.0D+01) + 19.*DLOG(TVI(1))
    A = DEXP(A)
    TE = DLOG(1. + A)
    TE = TVI(1)/(1. + TE*TVI(1)/85000.)
    TE2 = TE
C
576 IF (IX.EQ.1) GO TO 580
    IF ((IPSI.EQ.(NSR+1)).AND.(IX.EQ.(NSR+1))) GO TO 580
    IF (VAR(1).GE.VARI(IZTERM)) GO TO 580
    IF ((IPSI.LT.(NSR+1)).AND.(VAR(1).GE.DELX)) GO TO 580
    KIPF = KIPF + 1
    IF (KIPF.NE.IPF) GO TO 625
    KIPF = 0
580 SUMCI = 0.0
    DO 590 ISUM = 3,IMAXP2
        IF (VAR(ISUM).LT.0.0) NEG = -1
        IF (VAR(ISUM).LT.0.0) WRITE(15,585)
585 FORMAT(5X,'----- NEGATIVE CI -----'/)
590 SUMCI = SUMCI + VAR(ISUM)
    DO 595 ICM = 1,IMAX
595 CM(ICM) = VAR(ICM+2)*MU/MUI(ICM)
C                                     PRINT ANSWERS
    WRITE(8,600) IPSI,IX,IQ3,MODEL,XVAR,H,MU,PFTL,RHO,U,T,E,SPEC,SUMCI
600 FORMAT(1X,'PSI = ',I4,3X,'IX = ',I4,8X,'SHOCK J COND. = ',I2,14X,
    *'CVD MODEL = ',I2,/,1X,'X = ',D9.4,2X,'H = ',D9.4,2X,'MU = ',
    *E9.4,2X,'P = ',E9.4,3X,'RHO = ',E9.4,/,1X,'U = ',E9.4,2X,'T = ',
    *E9.4,2X,'E = ',E9.4,2X,'CIT = ',E9.4,2X,'SUMCI = ',E9.4/)
    IF (IQ7.EQ.IPSI) THEN
        WRITE(15,601) XVAR,T,CM(2),CM(4),SPEC
601 FORMAT(1X,E9.4,2X,E9.4,2X,E9.4,2X,E9.4,2X,E9.4)
    ENDIF
    DO 603 I = 3,IMAXP2
        J = I - 2
603 WRITE(8,605) J,VAR(I),J,CM(J),J,VAR(IMAX+I)
605 FORMAT(6X,'CI(',I2,') = ',D11.4,4X,'CM(',I2,') = ',E11.4,4X,
    * 'EVI(',I2,') = ',D11.4)
    WRITE(8,620)

```

```

620 FORMAT(/)
DO 621 I = 1,IMAX
IF (EVI(I).EQ.O.O) TVI(I) = O.O
IF (I.EQ.IMAX) THEN
WRITE(8,623) I,TVI(I),TE1,TE2
GO TO 621
ENDIF
WRITE(8,622) I,TVI(I)
621 CONTINUE
622 FORMAT(5X,'TVI(',I2,') = ',E11.4)
623 FORMAT(5X,'TVI(',I2,') = ',E11.4,4X,'TE1      = ',E11.4,4X,
*      'TE2      = ',E11.4)
WRITE(8,620)
IF (NEG.EQ.-1) WRITE(15,630)

C                                STORE INFO FOR DISSPLA
625 IF (IPSI.EQ.IS7) THEN
DO 626 ICM = 1,IMAX
626 CM(ICM) = VAR(ICM+2)*MU/MUI(ICM)
WRITE(70,*) (CM(I),I=1,IMAX)
WRITE(71,*) T,TVI(1),XVAR
ENDIF

C
630 FORMAT(1X,' A NEGATIVE CONCENTRATION , CI IN MAIN : STOP 670'/)
IF (NEG.EQ.-1) STOP

C                                STOP 670
C WHEN IPSI = (NSR+1), DELX AND STAGNATION STREAMLINES WILL BE COMPUTED
C                                IUNEG = 1 IF U**2 NEG. IN BASIC
IF (IUNEG.EQ.1) GO TO 735
IF ((IPSI.LT.(NSR+1)).AND.(VAR(1).LT.XPST(1))) GO TO 505
IF ((ISTAG.EQ.1).OR.(ISTAG.EQ.2)) GO TO 690
C                                ISTAG = 0 UNTIL AFTER EXTRAPOLATION FOR PSI = O.O STREAMLINE
C                                ISTAG = 1 FOR PSI = O.O STREAMLINE
C                                ISTAG = 2 FOR DELX STREAMLINE AND THEREAFTER
635 IG = IG+1
IGC = 0
IGE = IMAX + 2
PSIG(IG) = PSIS
DO 640 IG1 = 3,IMAXP2
IGC = IGC + 1
CIG(IGC,IG) = VAR(IG1)
IGE = IGE + 1
EIG(IGC,IG) = VAR(IGE)
640 CONTINUE
EG(IG) = E
TG(IG) = T
HG(IG) = H
IF (IPSI.LT.(NSR+1)) WRITE(8,725)
IF (IPSI.LT.(NSR+1)) GO TO 730
C                                IPSI = (NSR+1) IS THE STAGNATION STREAMLINE ;
C                                AT DELX ,EXTRAPOLATE FOR PSI = O.O VALUES
PSI = O.O
MG = 1
CALL FTLUP (PSI,E,MG,IG,PSIG,EG)
CALL FTLUP (PSI,T,MG,IG,PSIG,TG)
CALL FTLUP (PSI,H,MG,IG,PSIG,HG)
DO 650 IG2 = 1,IMAX
DO 645 IG1 = 1,NSR1
CIT(IG1) = CIG(IG2,IG1)
645 EIT(IG1) = EIG(IG2,IG1)
CALL FTLUP (PSI,CI(IG2),MG,IG,PSIG,CIT)
IF (CI(IG2).LT.O.O) CI(IG2) = 1.OE-08
CALL FTLUP (PSI,EVI(IG2),MG,IG,PSIG,EIT)
650 VAR(2+IMAX+IG2) = EVI(IG2)

C                                INITIALIZE FOR PSI = O.O STREAMLINE
CJ = XI
SPEC = O.O
II = 0
KEYINT = 0
IX = 0
SUM = O.O
DO 660 I = 1,IMAX
SUM = SUM + CI(I)/MUI(I)
IF (EVI(I).EQ.O.O) TVI(I) = O.O
IF (EVI(I).EQ.O.O) GO TO 660
XL = ((DGNI(I)*R*THETAI(I))/(MUI(I)*EVI(I)) + 1.O)
ALN = DLOG(XL)
TVI(I) = THETAI(I)/ALN
660 CONTINUE
MU = 1.O/SUM
U = DSQRT(2.O*(HSTAG - H))

```



```

      VAR(1) = DELX
C      COMPUTE P AND DPDX AT IPSI = (NSR+1) (AT THE BODY)
      IF (IANS.EQ.0) WRITE(8,665)
      IF ((IPSI.EQ.IS1).OR.(IPSI.EQ.IS2).OR.(IPSI.EQ.IS3)) WRITE(8,665)
      IF ((IPSI.EQ.IS4).OR.(IPSI.EQ.IS5).OR.(IPSI.EQ.IS6)) WRITE(8,665)
665  FORMAT(1X,'-- PSI = 0.000 STREAMLINE --'/)
      P(NSR+1) = PS + US/(RCM(NSR+1)*RSM(NSR+1))*(-PSIS)
      RHO = P(NSR+1)/(H - E)
      NSR2 = NSR + 2
      DO 670 I = NSR2, IZTERM
      READ(10,*) IXT, TS, ES, PS, RHOS, US, PSIS, HS
670  P(I) = PS + US/(RCM(I)*RSM(I))*(-PSIS)
      REWIND 10
      REWIND 11
      DPDX(NSR+1) = (-3.*P(NSR1)+4.*P(NSR2)-P(NSR3))/(2.*DELX)
C      EVALUATE DPDX FROM (NSR+2) TO IZTERM
      DO 680 I = NSR2, IZTERM
      IF (I.LT.IZTERM) THEN
        DPDX(I) = (-P(I-1) + P(I+1))/(2.*DELX)
      ELSE
        DPDX(I) = (P(I-2) - 4.*P(I-1) + 3.*P(I))/(2.*DELX)
      ENDIF
680  CONTINUE
      DO 685 I = 1, NSR
      READ(10,*) IX, TS, ES, PS, RHOS, US, PSIS, HS
685  READ(11,*) (EVIS(J), J = 1, IMAX)
      ISTAG = 1
      PSIS = 0.0
      GO TO 500
C      AT EACH X WHERE PHYSICAL SPACE CALCULATIONS ARE DESIRED SAVE
C      1./((RHO*U) ON EACH STREAMLINE
690  IF (XPST(LPS2).EQ.XPST(1)) ICNT = IPSI
      JJJ = IPSI - ICNT + 1
      IF (VAR(1).LT.XPST(LPS2)) GO TO 720
      IF (VAR(1).GT.XPST(LPS2)) GO TO 705
C      VAR(1) = XPST(LPS2)
      RHOURL = 1.0/(RHO*U)
      TEMP(LPS2, JJJ) = TVI(1)
      TEMPO(LPS2, JJJ) = T
      DO 691 III = 1, IMAX
691  RCON(LPS2, III, JJJ) = VAR(III+2)
      RRHO(LPS2, JJJ) = RHO
      WRITE(14,*) IPSI, RHOURL, VAR(1), PSIS
      GO TO 715
705  YPSIS = PREPSI + (XPST(LPS2) - PREX)*((PSIS - PREPSI)/
      * (VAR(1) - PREX))
      RHOURL = PRERU + (XPST(LPS2) - PREX)*((RHO*U - PRERU)/
      * (VAR(1) - PREX))
      TEMP(LPS2, JJJ) = PRETMP + (XPST(LPS2)-PREX)*((TVI(1)-PRETMP)/
      * (VAR(1) - PREX))
      TEMPO(LPS2, JJJ) = PRET + (XPST(LPS2)-PRET)*((T-PRET)/
      * (VAR(1)-PRET))
      DO 706 III = 1, IMAX
706  RCON(LPS2, III, JJJ) = PREC(III)+(XPST(LPS2)-PREX)*((VAR(III+2)
      * -PREC(III))/(VAR(1)-PREX))
      RRHO(LPS2, JJJ) = PRERHO + (XPST(LPS2)-PREX)*((RHO-PRERHO)/
      * (VAR(1) - PREX))
      RHOURL = 1.0/RHOURL
      WRITE(14,*) IPSI, RHOURL, XPST(LPS2), YPSIS
715  LPS2 = LPS2 + 1
      KSTAG = KSTAG + 1
720  PREX = VAR(1)
      PRERU = RHO*U
      PREPSI = PSIS
      PRETMP = TVI(1)
      PRET = T
      DO 721 III = 1, IMAX
721  PREC(III) = VAR(III+2)
      PRERHO = RHO
      IF (VAR(1).LT.VARI(IZTERM)) GO TO 505
C      END OF STREAMLINE
      WRITE(8,725)
725  FORMAT(1X,'XXXXXXXXXXXXXXXXXX'/)
      IF ((IPSI.EQ.(NSR+1)).AND.(ISTAG.EQ.1)) GO TO 425
730  GO TO 420
C      COMPUTE PHYSICAL SPACE VALUES
735  REWIND 10
      REWIND 11
      REWIND 12
      REWIND 13

```

```

WRITE(8,740) KSTAG
740 FORMAT(1X,'BEGIN PHYSICAL SPACE CALCULATIONS' KSTAG = ',I5/)
C
LPS3 = 0
IXSTR7 = 0
DO 840 LPS = 1,NXPST
REWIND 14
IK = 0
745 LPS3 = LPS3 + 1
IF (LPS3.GT.IZTERM) GO TO 840
IF (DABS(X1(LPS3) - XPST(LPS)).GT.1.OE-08) GO TO 745
DO 780 IPS = 1,KSTAG
READ(14,*) IPSI,RHOVRT,VAR(1),PSIS
VARMX = VAR(1) - XPST(LPS)
IF (DABS(VARMX).GT.1.OE-08) GO TO 780
IK = IK + 1
IBOB = IK - 1
IF (IK.GT.200) WRITE(15,770) IK
770 FORMAT(1X,'IK.GT.200,CHANGE DIMENSION OF RUT AND PSISTG,IK = ',I3)
RUT(IK) = RHOVRT
PSISTG(IK) = PSIS
780 CONTINUE
C
FIND SMALLEST DELTA PSI
SMALL = 200.0
DO 785 I = 2,IK
IF ((PSISTG(I) - PSISTG(I-1)).LT.SMALL) SMALL = PSISTG(I)
* - PSISTG(I-1)
785 CONTINUE
XDEL = PSISTG(IK)/SMALL
C
TRUNCATE
L = INT(XDEL + 1.0)
C
REMAINDERING M = 0 FOR EVEN, M = 1 FOR ODD
M = MOD(L,2)
C
MAKE L EVEN
IF (M.NE.0) L = L + 1
C
FIND INTEGRAL 1./(RHO*U) DELTA PSI FROM BODY TO SHOCK
USING SIMPSONS RULE ; L INCREMENTS , L+1 POINTS
C
FL = L
DPSIS = PSISTG(IK)/FL
PSII = 0.0
RB = RUT(1) + RUT(IK)
DO 800 I = 2,L,2
PSII = PSII + DPSIS
CALL FTLUP (PSII,RU1,1,IK,PSISTG,RUT)
IF (I.EQ.L) RB = RB + 4.0*RU1
IF (I.EQ.L) GO TO 800
PSII = PSII + DPSIS
CALL FTLUP (PSII,RU2,1,IK,PSISTG,RUT)
RB = RB + 4.0*RU1 + 2.0*RU2
800 CONTINUE
RB = RB*DPSIS/3.0
ARR = DSQRT(RSM(LPS3)**2 - 2.0*COSTM(LPS3)*RB)
YI = (RSM(LPS3) - ARR)/COSTM(LPS3)
YBDY(IBOB)=YI
ZE = ZSM(LPS3) + YI*SINTM(LPS3)
WRITE(8,815) PSISTG(1),X1(LPS3),ARR,YI,ZE
815 FORMAT(1X,'PSI = ',E12.5,/,1X,'X = ',E13.6,3X,'R = ',E13.6
* ,3X,'Y = ',E13.6,3X,'Z = ',E13.6/)
C
STORE BODY PTS. FOR DISSPLA
WRITE(72,*) ZE,ARR
C
DO 830 I = 2,IK
DPSI = PSISTG(I) - PSISTG(I-1)
C
TRAPEZOIDAL RULE
TR = (DPSI/2.0)*(RUT(I) + RUT(I-1))
RB = RB - TR
ARR = DSQRT(RSM(LPS3)**2 - 2.0*COSTM(LPS3)*RB)
YI = (RSM(LPS3) - ARR)/COSTM(LPS3)
Y(IBOB,I-1) = YI
ZE = ZSM(LPS3) + YI*SINTM(LPS3)
WRITE(8,815) PSISTG(I),X1(LPS3),ARR,YI,ZE
C
STORE STRLNE AND SHOCK COORD. FOR DISSPLA
IF (I.EQ.IK) THEN
IF ((I+NSRM1).EQ.IS7) THEN
IXSTR7 = IXSTR7 + 1
WRITE(73,*) ZSM(IS7),RSM(IS7),X1(IS7)
XSTR1(IXSTR7) = X1(IS7)
ZSTR1(IXSTR7) = ZSM(IS7)
RSTR1(IXSTR7) = RSM(IS7)
ENDIF

```

```

WRITE(74,*) ZE,ARR
GO TO 830
ENDIF
IF ((I+NSRM1).EQ.IS7) THEN
  IXSTR7 = IXSTR7 + 1
  WRITE(73,*) ZE,ARR,X1(LPS3)
  XSTR1(IXSTR7) = X1(LPS3)
  ZSTR1(IXSTR7) = ZE
  RSTR1(IXSTR7) = ARR
ENDIF

```

```

C
830 CONTINUE
  WRITE(8,835)
835 FORMAT(//)
840 CONTINUE

```

```

C
                                INTERPOLATE X COORD. TO DETERMINE (Z,R) AND XS

```

```

  STRL1 = 0.0
  PRER11 = 0.0
  PREZ11 = 0.0
  PRESTL = 0.0
  REWIND 71
841 READ(71,*,END=842) T11,TVN21,XS1
  CALL FTLUP (XS1,R11,2,IXSTR7,XSTR1,RSTR1)
  CALL FTLUP (XS1,Z11,2,IXSTR7,XSTR1,ZSTR1)
  SEG = DSQRT((PRER11 - R11)**2 + (PREZ11 - Z11)**2)
  IF (XS1.EQ.X1(IS7)) SEG = 0.0
  STRL1 = PRESTL + SEG
  WRITE(75,*) XS1,Z11,R11,STRL1
  PRESTL = STRL1
  PRER11 = R11
  PREZ11 = Z11
  GO TO 841

```

```

C
                                RADIATION MODELS

```

```

842 IF (IQ4.EQ.0) GO TO 845
  IF (IQ5.EQ.0) WRITE(8,9191)
9191 FORMAT(///,' NONEQUILIBRIUM RADIATION MODELS',///)
  IF (IQ4.EQ.1) CALL OLRAD (IQ4,IQ5,IQ6,IQ8,IQ9,IQ10)
  IF (IQ4.EQ.2) CALL CARRAD (IQ4,IQ5,IQ9)
  IF (IQ4.EQ.3) CALL CORRAD (IQ4,IQ5,IQ6,IQ8,IQ9)
  IF (IQ4.EQ.4) CALL ANDRAD (IQ4,IQ5,IQ9)
  IF ((IQ4.EQ.5).OR.(IQ4.EQ.6)) THEN
    CALL OLRAD (IQ4,IQ5,IQ6,IQ8,IQ9,IQ10)
    CALL CARRAD (IQ4,IQ5,IQ9)
    CALL CORRAD (IQ4,IQ5,IQ6,IQ8,IQ9)
    CALL ANDRAD (IQ4,IQ5,IQ9)
  ENDIF
  IF (IQ5.EQ.0) THEN
    DO 1920 K = 3,NXCON
      WRITE(8,1911) XPST(K)
    DO 1920 J = 1,NPTS(K)
      WRITE(8,1921) J,ALPN(K,J),ALPO(K,J),BETN(K,J),BETO(K,J)
1921  *  FORMAT('  J = ',I2,'  An = ',D11.4,'  Ao = ',D11.4/
      ,', 2X,'  BN = ',D11.4,'  BO = ',D11.4)
1920  CONTINUE
    ENDIF
    DO 1912 K = 3,NXCON
      WRITE(8,1911) XPST(K)
1911  *  FORMAT(///,' AT X = ',D11.4,///)
    DO 1912 J = 1,NPTS(K)
      WRITE(8,1913) J,TEMP(K,J),RRHO(K,J),TEMPO(K,J)
1913  *  FORMAT('  J = ',I2,'  TE = ',F10.2,'  RHO = ',D11.4,'  T = ',F10.2)
1912  CONTINUE
    DO 1918 K = 3,NXCON
      WRITE(8,1911) XPST(K)
    DO 1918 J = 1,NPTS(K)
      WRITE(8,1914) J
1914  *  FORMAT(/,' AT J = ',I2,'  NUMBER DENSITIES ARE ',/)
    DO 1918 I = 1,IMAX
      WRITE(8,1915) I,RCON(K,I,J)
1915  *  FORMAT('  SPECIES = ',I2,'  N = ',D11.4)
1918  CONTINUE
    WRITE(8,835)
845 WRITE(8,850)
850 FORMAT(1X,'END THIS CASE , HALLELUJAH '/')
    REWIND 8
    REWIND 14
    STOP
  END

```

C

C-----

```

C
SUBROUTINE BASIC
C
C BASIC CALLED BY CALITH TO EVALUATE DERIVATIVES H DOT,
C SUB I DOT AND EV SUB I DOT ; DERIVATIVES START IN DER(1)
C
IMPLICIT REAL*8(A-H,O-Z)
COMMON /A1/ TE1,TE2,M,IX,IMAX,IPSI,MODEL
COMMON /A2/ P(500),DPDX(500),VARI(500)
COMMON /A3/ EVIINF(25),THETAI(25),MUI(25),FI(25),DELHI(25),
* CIINF(25),LI(25)
COMMON /A4/ R,MUINF,DELTA,LAMSQ,OMEGSQ,OMEGA,MU,T,U,TINF
COMMON /A5/ EPSIIL(20,25),GIL(20,25),CID(200,50)
COMMON /A6/ VAR(52),CUVAR(52),DER(51)
COMMON /A7/ MJ(50),EJ(50),AJ(50),BJ(50),DIRECT(50)
COMMON /A8/ TVI(25),DGENI(25),BI(25),NI(25)
COMMON /A10/ NUIJ(26,50),AIJ(25,50),NUIPJ(25,50)
COMMON /A11/ SIGIK(25,25),ALPIK(25,25),BETAIK(25,25)
COMMON /A13/ SP,TS,DELX,ZSTERM,IZTERM,NSR,MW
COMMON /A14/ EINF,PINF,RHOINF,VINF,E,JMAX,KEYINT,RHO,HSTAG
COMMON /A15/ PFTL,KITR1,NIP(25),UP(25)
COMMON /A16/ WE(25),WEZE(25),WEYE(25),WEZE(25),IUNEG
COMMON /A18/ ITNEG,IEXP
DIMENSION PHI(50),SJ(50),DCIDX(50),EVP(100)
DIMENSION CI(25),EVI(25),DCIDX(25),DEVIDX(25),EVIBAR(25)
EQUIVALENCE (CUVAR(2),H),(CUVAR(3),CI(1))
EQUIVALENCE (DER(1),DHDX),(DER(2),DCIDX(1))
REAL*8 KPJ,KEQ,KJ(50),LAMSQ,MU,MUI,MUIOT,MUINF,ND
C
C DER(1) = DHDX MUST BE DHDX AS H MAY BE + OR -
C DER(2) = DCIDX(1)
C
C DER(1+IMAX) = DCIDX(IMAX)
C DER(1+IMAX+1) = DEVIDX(1)
C DER(1+IMAX+IMAX) = DEVIDX(IMAX)
C
C CUVAR(1) = X
C CUVAR(2) = H, ENTHALPY - IT MUST BE H WHICH MAY BE + OR -
C CUVAR(3) = CI(1)
C
C CUVAR(2+IMAX) = CI(IMAX)
C CUVAR(2+IMAX+1) = EVI(1)
C CUVAR(2+IMAX+IMAX) = EVI(IMAX)
C
C KEYINT = 1 AT END OF 1ST INTERVAL
C IF (ITNEG.EQ.1.OR.IEXP.EQ.1) RETURN
C IF (M.NE.O) GO TO 10
C DO 5 I = 1,IMAX
C J1 = I + IMAX + 2
C 5 CUVAR(J1) = VAR(J1)
C INTERPOLATE FOR P ACROSS INTEGRATION INTERVAL
C 10 DO 15 I = 1,IMAX
C 15 EVI(I) = CUVAR(2+IMAX+I)
C MFTL = 1
C NFTL = IZTERM - IPSI + 1
C CALL FTLUP (CUVAR(1),PFTL,MFTL,NFTL,VARI(IPSI),P(IPSI))
C INTERPOLATE FOR DPDX
C CALL FTLUP (CUVAR(1),DPFTL,MFTL,NFTL,VARI(IPSI),DPDX(IPSI))
C IF (KEYINT.EQ.O) GO TO 95
C SUM = O.O
C DO 45 I = 1,IMAX
C USE SHOCK VALUES FOR INITIAL COMPUTATION ON EACH STREAMLINE
C SUM = SUM + CI(I)/MUI(I)
C IF (EVI(I).EQ.O.O) TVI(I) = O.O
C IF (EVI(I).EQ.O.O) GO TO 45
C XL = ((DGENI(I)*R*THETAI(I))/(MUI(I)*EVI(I)) + 1.O)
C ALN = DLOG(XL)
C COMPUTE TVI
C TVI(I) = THETAI(I)/ALN
C 45 CONTINUE
C MU = 1.O/SUM
C U2 = 2.O*(HSTAG - H)
C IF (U2.LT.O.O) WRITE(15,50) H
C 50 FORMAT(1X,'U**2 IS NEG IN BASIC , H = ',E11.4,2X,
C * 'END STREAMLINE INTEGRATION')
C IF (U2.LT.O.O) IUNEG = 1
C IF (U2.LT.O.O) RETURN
C U = DSQRT(U2)
C ITERATE FOR E
C KCODE = O

```

```

55 KITR1 = 0
   ICODE = 0
   MAXI = 50
   TOL1 = .001
   TOL2 = .00001
C
   CALL FOFE (MAXI,TOL1,TOL2,ICODE,E)
C
   IF (ICODE.EQ.0) GO TO 75
   IF (ICODE.EQ.1) WRITE(15,60)
60  FORMAT(1X,'* MAXIMUM ITERATION EXCEEDED IN SUB BASIC *: STOP 66'//)
   IF (ICODE.EQ.2) WRITE(15,65)
65  FORMAT(1X,'*** DERIVATIVE = 0 IN SUB BASIC **** : STOP 66'//)
   WRITE(15,70) ICODE
70  FORMAT(1X,'ICODE = ',I2,' IN SUB BASIC '//)
   IF (ICODE.NE.1) STOP
   IF (KCODE.EQ.3) STOP
C
   WHEN ICODE = 1, TRY A NEW STARTING E. DO THIS 2 TIMES ;   STOP 66
   KCODE = KCODE + 1
   GO TO 55
75  IF (IEXP.NE.1) GO TO 85
   KCODE = KCODE + 1
   IEXP = 0
   IF (KCODE.EQ.1) GO TO 55
   IF (KCODE.GT.2) RETURN
   E = H - H*1.E-05
   GO TO 55
85  RHO = PFTL/(H - E)
   T = (PFTL*MU)/(RHO*R)
   IF (T.GT.0.0) GO TO 95
   WRITE(15,90) T,RHO,PFTL,H,E,MU
90  FORMAT(1X,'T NEGATIVE = ',E12.4,1X,'RHO = ',E11.4,1X,'PFTL = ',
*      E11.4,1X,'H = ',E11.4,1X,'E = ',E11.4,1X,'MU = ',E11.4,2X,
*      'IN SUB BASIC'//)
   E = H - H*1.OE-06
   ITNEG = 1
   RETURN
95  KEYINT = 1
   PI = 4.0*ATAN(1.0).
   BC = 1.38054E-16
   AVGN = 6.02252E+23
C
   COMPUTE DCIDX AND DEVIDX FOR EACH SPECIE I
   DO 210 I = 1,IMAX
   DCIDX(I) = 0.0
   CVSUM1 = 0.0
   CVSUM2 = 0.0
C
   CORRECTED RELAXATION TIME FOR PARK MODEL
   IF (MODEL.EQ.4.AND.I.EQ.1) THEN
   ND = RHO*AVGN*CI(I)/MUI(I)
   SIGMAV = 10.0E-17*(50000./T)**2
   C = DSQRT(8.*AVGN*BC*T/(PI*MUI(I)))
   TAUC = 1.0/(C*SIGMAV*ND)
   ENDIF
C
   FOR EACH REACTION J
   DO 190 J = 1,JMAX
   ICT = MOD(J,2)
C
   II IS THE SELECTED SPECIE FOR REACTION J
   KNT = 0
   IF (ICT.EQ.1) THEN
   IF (DIRECT(J).EQ.1.0) PHIC = 1.0
   IF (DIRECT(J).EQ.2.0) PHIC = 2.0
   II = MJ(J)
100  IF (M.EQ.0) GO TO 115
   IF (FI(II).NE.0.0) GO TO 120
115  IF (PHIC.EQ.1.0) PHI(J) = 1.0
   IF (PHIC.EQ.2.0) PHI(J+1) = 1.0
   GO TO 125
120  IF (MODEL.EQ.3.AND.UP(II).NE.0.0) GO TO 122
   TEM = DEXP(THETAI(II)/TVI(II))
   TEM1 = DEXP(THETAI(II)/T)
   TEM2 = DEXP(-NI(II)*(THETAI(II)/TVI(II) - THETAI(II)/T))
   TEM3 = DEXP(THETAI(II)/TVI(II) - THETAI(II)/T)
   IF (TEM3.EQ.1.0.AND.PHIC.EQ.1.0) PHI(J) = 1.0
   IF (TEM3.EQ.1.0.AND.PHIC.EQ.2.0) PHI(J+1) = 1.0
   IF (TEM3.EQ.1.0) GO TO 125
C
   PHI = COUPLING COEFF.; FOR CVD,CVDV AND PARK MODEL
   IF (PHIC.EQ.1.0) PHI(J) = ((1.0-TEM2)/(TEM3-1.0))*(TEM - 1.0)
*      /(TEM1 - 1.0)/NI(II)
   IF (PHIC.EQ.2.0) PHI(J+1) = ((1.0-TEM2)/(TEM3-1.0))*(TEM - 1.0)
*      /(TEM1 - 1.0)/NI(II)

```

```

GO TO 125
122 USUM = 0.0
    TSUM = 0.0
    TFSUM = 0.0
    TVTSUM = 0.0
    TF = 1.0/((1./TVI(II) - 1./T - 1./UP(II))
    NIC = NIP(II) + 1
    DO 123 N = 1,NIC
    XN = N
    EEVP = XN - 0.5
    EVP(N) = EEVP*(WE(II) + EEVP*(-WE(II) + EEVP*(WEYE(II) +
*      WEZE(II)*EEVP)))
    EVP(N) = EVP(N) - EVP(1)
    USUM = USUM + DEXP(EVP(N)/UP(II))
    TSUM = TSUM + DEXP(-EVP(N)/T)
    TFSUM = TFSUM + DEXP(-EVP(N)/TF)
123 TVTSUM = TVTSUM + DEXP(-EVP(N)/TVI(II))
C      COUPLING COEFF. FOR CVDV-PREFERENTIAL MODEL
    IF (PHIC.EQ.1.0) PHI(J) = (TSUM*TFSUM)/(USUM*TVTSUM)
    IF (PHIC.EQ.2.0) PHI(J+1) = (TSUM*TFSUM)/(USUM*TVTSUM)
125 KNT = KNT + 1
    IF (KNT.EQ.1.AND.PHIC.EQ.1.0) THEN
    PHIC = 2.0
    II = MJ(J+1)
    GO TO 100
    ENDIF
    IF (KNT.EQ.1.AND.PHIC.EQ.2.0) THEN
    PHIC = 1.0
    II = MJ(J+1)
    GO TO 100
    ENDIF
C      START OF LOOP TO DETERMINE EQUILIBRIUM CONSTANT
    BETAJ = 0.0
    FJORT = 0.0
    DO 130 K = 1,IMAX
    SUMG = 0.0
    LII = LI(K)
    DO 128 L = 1,LII
128 SUMG = SUMG + GIL(L,K)*DEXP(-EPSIIL(L,K)/T)
C      MUJOT = (CHEMICAL POTENTIAL)/T
    MUJOT = - (BI(K) + (5. + 2.*FI(K))/2.*(DLOG(TINF) +
*      DLOG(T/TINF)) + FI(K)*DLOG(1./(1. - DEXP(-THETAI(K)/T)))
*      + DLOG(SUMG/GIL(1,K))) + DELHI(K)/(R*T)
    BETAJ = BETAJ + (NUPIJ(K,J) - NUIJ(K,J))
130 FJORT = FJORT + (NUPIJ(K,J) - NUIJ(K,J))*MUJOT
C      KPJ = EQUIL. CONST. (IN TERMS OF PARTIAL PRESSURES)
    KPJ = DEXP(-FJORT)
C      KEQ = EQUILIBRIUM CONSTANT
    KEQ = KPJ*(9.8688225E-07*R*T)**(-BETAJ)
C      KJ = RATE CONSTANT (FORW: J = 1,3,.. BACK: J = 2,4,..)
    IF (DIRECT(J).EQ.1.0) THEN
    KJ(J) = AJ(J)*T**BJ(J)*DEXP(-EJ(J)/T)
    IF (J.EQ.19) KJ(J) = AJ(J)*TVI(1)**BJ(J)*DEXP(-EJ(J)/TVI(1))
    KJ(J+1) = KJ(J)/KEQ
    ELSE
    KJ(J+1) = AJ(J)*T**BJ(J)*DEXP(-EJ(J)/T)
    IF (J.EQ.19) KJ(J+1) = AJ(J)*TVI(1)**BJ(J)*DEXP(-EJ(J)/TVI(1))
    KJ(J) = KJ(J+1)*KEQ
    ENDIF
    ENDIF
    IF (NUIJ(IMAX+1,J).EQ.0) SJ(J) = 1.0
    IF (NUIJ(IMAX+1,J).EQ.0) GO TO 140
    SUM = 0.0
    DO 135 ISUM = 1,IMAX
135 SUM = SUM + AIJ(ISUM,J)*CI(ISUM)/MUI(ISUM)
    SJ(J) = RHO*SUM
140 PROD = 1.0
    DO 180 IPROD = 1,IMAX
    IF (NUIJ(IPROD,J).EQ.0) GO TO 180
    TEM = RHO*CI(IPROD)/MUI(IPROD)
    IF (TEM.GE.0.0) GO TO 165
C      LET PROBLEM COMPUTE PROD. WHEN CI NEG ; DECIDE IN MAIN ABOUT ACCEPTING NEG CI
    LABW = MOD(NUIJ(IPROD,J),2)
C      LABW = 0 FOR EVEN NUIJ, ABW = 1 FOR ODD NUIJ
    IF (LABW.EQ.0) PROD = PROD*(-TEM)**NUIJ(IPROD,J)
    IF (LABW.EQ.1) PROD = -1.0*PROD*(-TEM)**NUIJ(IPROD,J)
    GO TO 180
165 PROD = PROD*TEM**NUIJ(IPROD,J)
180 CONTINUE
    IF (ICT.EQ.1) PRODF = PROD

```

```

      BETAIJ = NUPIJ(I,J) - NUIJ(I,J)
      DCIJDJ(J) = PHI(J)*KJ(J)*SJ(J)*(MUI(I)/(RHO*U))*PROD*BETAIJ
C      DCIDJ = RATE OF PRODUCTION OF THE CONCENTRATION OF SPECIE I
      DCIDJ(I) = DCIDJ(I) + DCIJDJ(J)
      IF (ICT.EQ.O.AND.MODEL.GT.1) THEN
C          QIJ = NET RATE OF PRODUCTION
          QIJ = DCIJDJ(J-1) + DCIJDJ(J)
          DENOM = KEQ*PHI(J-1)*SJ(J-1)*PRODF
          IF (DENOM.EQ.O.O) GO TO 190
C          CHI = DEGREE OF NON-EQUILIBRIUM
          CHI = 1.0 - (PHI(J)*SJ(J)*PROD/DENOM)
          DENOM1 = CI(I)*CHI
          IF (DENOM1.EQ.O.O) GO TO 190
          CVSUM1 = CVSUM1 + QIJ/DENOM1
          CVSUM2 = CVSUM2 + QIJ*(1.0 - CHI)/DENOM1
      ENDIF
190 CONTINUE
C
      TEM = DEXP(THETAI(I)/T)
      TEM1 = (FI(I)*DGENI(I)*R*THETAI(I))/MUI(I)
      EVIBAR(I) = TEM1/(TEM - 1.0)
      IF (M.NE.O) GO TO 195
      VAR(I+IMAX+2) = EVIBAR(I)
      DEVIDX(I) = 0.0
      GO TO 210
195 TAUSUM = 0.0
      IF (FI(I).EQ.O.O) GO TO 210
C          CVD MODEL
          DO 200 K = 1,IMAX
          TEM3 = DEXP(-THETAI(I)/T)
          TEM4 = DEXP(SIGIK(K,I)*T**(-1./3.))
          TEM5 = (FI(I)*ALPIK(K,I))/PFTL
          TAUIK = TEM5*(T**BETAIK(K,I)*TEM4)/(1.0 - TEM3)
C          MILLIKAN & WHITE RELAXATION DATA
          IF (MW.EQ.1) THEN
          IF (I.EQ.1) TAUIK = 1.9E-05*DEXP(216.45*T**(-1./3.))/PFTL
          IF (I.EQ.8) TAUIK = 3.1845E-05*DEXP(198.66*T**(-1./3.))/PFTL
          ENDIF
          TEM6 = CI(K)*(EVIBAR(I) - EVI(I))
C          PARK MODEL
          IF (MODEL.EQ.4.AND.I.EQ.1) THEN
          TAUPL = TAUIK + TAUC
          PF = DABS((T - TVI(I))/(TS - TINF))**((SP - 1.))
          TAUSUM = TAUSUM + (TEM6/(TAUPL*U))*PF
          GO TO 200
          ENDIF
          TAUSUM = TAUSUM + TEM6/(TAUIK*U)
200 CONTINUE
          IF (MODEL.EQ.1) GO TO 207
          IF (MODEL.EQ.3) GO TO 202
C          ADDITIONAL TERMS FOR CVDV & PARK MODEL
          IF (I.EQ.5.OR.I.EQ.6.OR.I.EQ.8) GO TO 207
          TEM7 = DEXP(THETAI(I)/TVI(I) - THETAI(I)/T)
          TEM75 = DEXP(NI(I)/4.*(THETAI(I)/TVI(I) - THETAI(I)/T))
          TEM8 = TEM75**4
          CVDVT = ((THETAI(I)/(TEM7 - 1.) - NI(I)*THETAI(I)/(TEM8 - 1.))
          *R/MUI(I) - EVI(I))*CVSUM1 - ((0.5*(NI(I) - 1.)*THETAI(I)
          *R/MUI(I)) - EVI(I))*CVSUM2
          TAUSUM = TAUSUM + CVDVT
          GO TO 207
C          ADDITIONAL TERMS FOR CVDV-PREFERENTIAL MODEL
202 IF (I.EQ.5.OR.I.EQ.6.OR.I.EQ.8) GO TO 207
          USUM1 = 0.0
          USUM2 = 0.0
          TFSUM1 = 0.0
          TFSUM2 = 0.0
          NIC = NIP(I) + 1
          TF = 1.0/(1./TVI(I) - 1./T - 1./UP(I))
          DO 204 N = 1,NIC
          XN = N
          EEVP = XN - 0.5
          EVP(N) = EEVP*(WE(I) + EEVP*(-WE(XE(I) + EEVP*(WEYE(I) +
          * WEZE(I)*EEVP)))
          EVP(N) = EVP(N) - EVP(1)
          USUM1 = USUM1 + DEXP(EVP(N)/UP(I))
          USUM2 = USUM2 + EVP(N)*DEXP(EVP(N)/UP(I))
          TFSUM1 = TFSUM1 + DEXP(-EVP(N)/TF)
204 TFSUM2 = TFSUM2 + EVP(N)*DEXP(-EVP(N)/TF)
          GBAR = (USUM2/USUM1)*(R/MUI(I))
          EBAR = (TFSUM2/TFSUM1)*(R/MUI(I))

```

```

      CVDVPT = (EBAR - EVI(I))*CVSUM1 - (GBAR - EVI(I))*CVSUM2
      TAUSUM = TAUSUM + CVDVPT
C
      DEVIDX = EQUILIBRIUM VIBRATIONAL ENERGY
207 DEVIDX(I) = TAUSUM
      DER(1+IMAX+I) = DEVIDX(I)
210 CONTINUE
C
      COMPUTE DHDX
      DHDX = DPFTL/RHO
      RETURN
      END
C
C-----
C
      SUBROUTINE FOFE (MAXI,TOL1,TOL2,ICODE,E1)
C
C      CALLED BY SUB BASIC TO EVALUATE E BY NEWTON ITERATION TECHNIQUE
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /A1/ TE1,TE2,M,IX,IMAX,IPSI,MODEL
      COMMON /A2/ P(500),DPDX(500),VARI(500)
      COMMON /A3/ EVIINF(25),THETAI(25),MUI(25),FI(25),DELHI(25),
      *          CIINF(25),LI(25)
      COMMON /A4/ R,MUINF,DELTA,LAMSQ,OMEGSQ,OMEGA,MU,T,U,TINF
      COMMON /A5/ EPSIIL(20,25),GIL(20,25),CID(200,50)
      COMMON /A6/ VAR(52),CUVAR(52),DER(51)
      COMMON /A7/ MJ(50),EJ(50),AJ(50),BJ(50),DIRECT(50)
      COMMON /A8/ TVI(25),DGENI(25),BI(25),NI(25)
      COMMON /A10/ NUIJ(26,50),AIJ(25,50),NUIPJ(25,50)
      COMMON /A15/ PFTL,KITR1,NIP(25),UP(25)
      COMMON /A18/ ITNEG,IEXP
      DIMENSION CI(25)
      EQUIVALENCE (CUVAR(3),CI(1)),(CUVAR(2),H)
      REAL*8 MUI,LAMSQ,MUINF,MU
C
      IF (IEXP.EQ.1.OR.ITNEG.EQ.1) RETURN
      KITR1 = KITR1 + 1
      ITER = 0
      PORHO = H - E1
      T = PORHO*MU/R
1   ITER = ITER + 1
      E = 0.0
      DE = 0.0
      DO 35 I = 1,IMAX
      IF (TVI(I).EQ.0.0) THEN
        TEM1 = (1.5*R*T)/MUI(I) + (FI(I)*R*T)/MUI(I)
        DTEM1 = 1.5*R/MUI(I) + FI(I)*R/MUI(I)
        GO TO 15
      ENDIF
      TEM = DEXP(THETAI(I)/TVI(I))
      TEM1 = (1.5*R*T)/MUI(I) + (FI(I)*R*T)/MUI(I) +
      *      (FI(I)*R*THETAI(I))/(MUI(I)*(TEM - 1.0))
      DTEM1 = 1.5*R/MUI(I) + FI(I)*R/MUI(I) + (R*FI(I)*THETAI(I)**2)
      *      /(MUI(I)*TVI(I)**2)*(TEM/(TEM - 1.0)**2)
15  SUMG = 0.0
      SUMGE = 0.0
      DSUMGE = 0.0
      LII = LI(I)
C
      DO 25 L = 1,LII
      TEM3 = - EPSIIL(L,I)/T
      IF (TEM3.LT.741.67) GO TO 22
      IEXP = 1
      RETURN
22  TEM2 = DEXP(TEM3)
      SUMG = SUMG + GIL(L,I)*TEM2
      SUMGE = SUMGE + GIL(L,I)*EPSIIL(L,I)*TEM2
      IF (SUMGE.LT.1.0D+34) GO TO 25
      IEXP = 1
      RETURN
25  DSUMGE = DSUMGE + TEM2*GIL(L,I)*EPSIIL(L,I)**2
      DEVI = R/MUI(I)*(((SUMG*DSUMGE/T**2) - (SUMGE/T)**2)
      *      /SUMG**2)
      DEI = DTEM1 + DEVI
      EI = TEM1 + R/MUI(I)*(SUMGE/SUMG) + DELHI(I)/MUI(I)
      DE = DE + DEI*CI(I)
      E = E + EI*CI(I)
35  CONTINUE
C
      IF (E.GT.H) ITNEG = 1
      DFT = - DE

```



```

FT = E1 - E
IF (DFT.EQ.O.O) THEN
  ICODE = 2
  RETURN
ENDIF
TS1 = T - (FT/DFT)
DELT = DABS(TS1 - T)
T = TS1
E1 = H - R*T/MU
IF (ITER.GE.MAXI) THEN
  ICODE = 1
  RETURN
ENDIF
IF (DELT.GT.TOL1) THEN
  ITER1 = ITER
  GO TO 1
ENDIF
ITER2 = ITER1 + 10
IF (DELT.LE.TOL2.OR.ITER.GE.ITER2) RETURN
GO TO 1
END

```

```

C
C-----
C
SUBROUTINE FOFTS1 (MAXI,TOL1,TOL2,ICODE,TSG)
C
C CALLED BY MAIN TO EVALUATE TS BY NEWTON ITERATION TECHNIQUE
C

```

```

  IMPLICIT REAL*8(A-H,O-Z)
  COMMON /A1/ TE1,TE2,M,IX,IMAX,IPSI,MODEL
  COMMON /A2/ P(500),DPDX(500),VARI(500)
  COMMON /A3/ EVIINF(25),THETAI(25),MUI(25),FI(25),DELHI(25)
  * ,CIINF(25),LI(25)
  COMMON /A4/ R,MUINF,DELTA,LAMSQ,OMEGSQ,OMEGA,MU,T,U,TINF
  COMMON /A5/ EPSIIL(20,25),GIL(20,25),CID(200,50)
  COMMON /A6/ VAR(52),CUVAR(52),DER(51)
  REAL*8 MUI,LAMSQ,MUINF,MU

```

```

C
  ITER = 0
1  ITER = ITER + 1
  ES = 0.0
  DES = 0.0
  DO 25 I = 1,IMAX
    IF (M.EQ.1) EVIS = EVIINF(I)
    IF (M.EQ.1) GO TO 5
    TEM = DEXP(THETAI(I)/TSG)
    EVIS = (R*THETAI(I))/(MUI(I)*(TEM-1.0))*FI(I)
    DEVIS = (R*FI(I)*THETAI(I)**2)/(MUI(I)*TSG**2)*(TEM/(TEM-1.0)**2)
5  SUMG = 0.0
  SUMGE = 0.0
  DSUMGE = 0.0
  LII = LI(I)

```

```

C
  DO 10 L = 1,LII
    TEM1 = DEXP(- EPSIIL(L,I)/TSG)
    SUMG = SUMG + TEM1*GIL(L,I)
    SUMGE = SUMGE + TEM1*GIL(L,I)*EPSIIL(L,I)
10  DSUMGE = DSUMGE + TEM1*GIL(L,I)*EPSIIL(L,I)**2
    EEIS = (R/MUI(I))*(SUMGE/SUMG)
    DEEIS = R/MUI(I)*(((SUMG*DSUMGE/TSG**2) - (SUMGE/TSG)**2)
    * /SUMG**2)
    EIS = (1.5*R*TSG)/MUI(I) + (FI(I)*R*TSG)/MUI(I) + EVIS + EEIS
    * + DELHI(I)/MUI(I)
    DEIS = 1.5*R/MUI(I) + FI(I)*R/MUI(I) + DEVIS + DEEIS
    ES = ES + EIS*CIINF(I)
    DES = DES + DEIS*CIINF(I)
25  CONTINUE

```

```

C
  FAC = DSQRT(OMEGSQ - 2.0*LAMSQ*(DELTA - ES))
  FIN = (2.0*MUINF*(DELTA - ES)*FAC)/(R*(OMEGA + FAC))
  FTS = TSG - FIN
  FAC1 = R*(OMEGA+FAC)*(MUINF*(DELTA-ES)*(2.0/FAC)*LAMSQ*DES -
  * 2.0*FAC*MUINF*DES) - 2.0*MUINF*(DELTA-ES)*R*LAMSQ*DES
  FAC2 = R**2*(OMEGA + FAC)**2
  DFTS = 1.0 - FAC1/FAC2
  IF (DFTS.EQ.O.O) THEN
    ICODE = 2
    RETURN
  ENDIF
  TSG1 = TSG - (FTS/DFTS)

```

```

DELTS = DABS(TSG1 - TSG)
TSG = TSG1
IF (ITER.GE.MAXI) THEN
  ICODE = 1
  RETURN
ENDIF
IF (DELTS.GT.TOL1) THEN
  ITER1 = ITER
  GO TO 1
ENDIF
ITER2 = ITER1 + 10
IF (DELTS.LE.TOL2.OR.ITER.GE.ITER2) RETURN
GO TO 1
END

```

```

C
C-----
C
C      SUBROUTINE CHECK
C
C      CHECK CALLED BY CALITH TO MAKE DECISION TO ACCEPT ANSWERS
C
C      IF ACCEPTABLE      ; SET ELB = 0 AND RETURN
C      IF NOT ACCEPTABLE ; MODIFY SPEC AND CI ; SET ELB = 1.0 AND RETURN
C
C      IMPLICIT REAL*8(A-H,O-Z)
C      COMMON /A1/ TE1,TE2,M,IX,IMAX,IPSI,MODEL
C      COMMON /A2/ P(500),DPDX(500),VARI(500)
C      COMMON /A3/ EVIINF(25),THETA1(25),MUI(25),FI(25),DELHI(25),
C      *      CIINF(25),LI(25)
C      COMMON /A4/ R,MUINF,DELTA,LAMSQ,OMEGSQ,OMEGA,MU,T,U,TINF
C      COMMON /A5/ EPSIIL(20,25),GIL(20,25),CID(200,50)
C      COMMON /A6/ VAR(52),CUVAR(52),DER(51)
C      COMMON /A13/ SP,TS,DELX,ZSTERM,IZTERM,NSR,MW
C      COMMON /A17/ ELB,SPEC,CJ,TPREV,HPREV,HCHECK,TCHECK
C      COMMON /A18/ ITNEG,IEXP
C      REAL*8 MUI,MU,MUINF,LAMSQ
C      EQUIVALENCE (CUVAR(2),H)
C
C      IF (IEXP.NE.1) GO TO 1
C      GO TO 15
C
C      REDUCE COMPUTING INTERVAL TO TRY AND AVOID EXP ERROR STOP
C      1 IF (ITNEG.EQ.0) GO TO 4
C      E.GT.H SOMETIMES LEADS TO NEG T--REDUCE INTERVAL TO TRY TO AVOID INSTABILITY
C      GO TO 15
C      4 IMAXP1 = IMAX + 1
C      DO 10 I = 3,IMAXP1
C
C      CUVAR(2) = H MAY BE NEGATIVE
C      IF (CUVAR(I).LT.0.0) WRITE(15,5) I,IX,CUVAR(I)
C      IF (CUVAR(I).LT.0.0) GO TO 15
C      5 FORMAT(1X,'NEG CI AT I = ',I3,1X,'IX = ',I3,1X,'CUVAR(I) = ',
C      *      E12.5,1X,'IN SUB CHECK'//)
C      10 CONTINUE
C      IF (IX.LT.3) GO TO 30
C      IF ((IPSI.EQ.(NSR+1)).AND.(IX.LT.(NSR+3))) GO TO 30
C      IF (DABS(TPREV - T)/T.LT.TCHECK) GO TO 25
C      WRITE(15,11)
C      11 FORMAT(1X,'CHECK ABS(TPREV-T)/T.GE.TCHECKT IN SUB CHECK'//)
C      15 IF (SPEC.GT.1.0E-15) GO TO 20
C      WRITE(15,*)'SPEC .LT. 1.0E-15 IN SUB CHECK'
C      STOP
C
C      STOP 30
C
C      REDUCE INTERVAL
C
C      20 SPEC = SPEC/4.0
C      CJ = SPEC
C      ELB = 1.0
C      WRITE(15,22) SPEC,IPSI,IX,T,TPREV,H,HPREV,IEXP,ITNEG
C      22 FORMAT(1X,'REDUCED SPEC = ',E9.4,2X,'IPSI = ',I3,2X,'IX = ',I3,
C      *      2X,'T = ',E9.4,2X,'TPREV = ',E9.4,/,1X,'H = ',E9.4,3X,
C      *      'HPREV = ',E9.4,3X,'IEXP = ',I3,3X,'ITNEG = ',I3,' IN CHECK'//)
C      ITNEG = 0
C      IEXP = 0
C      RETURN
C      25 IF (DABS((HPREV - H)/H).GT.HCHECK) WRITE(15,27)
C      27 FORMAT(1X,'CHECK ABS((HPREV-H)/H).GT.HCHECKT IN SUB CHECK'//)
C      IF (DABS((HPREV - H)/H).GT.HCHECK) GO TO 15
C
C      ACCEPTABLE
C
C      30 TPREV = T
C      HPREV = H
C      ELB = 0.0
C      RETURN

```

END

```
C
C-----
C
C      SUBROUTINE SHOCKG (NXPSTM,ITK)
C
C      SUBROUTINE SHOCKG CALLED BY MAIN AND MUST BE SUPPLIED BY USER.
C      X,ZS,RS,RC,COST,SINT FOR EACH X FROM 0.0 TO X AT ZSTERM IN INCREMENTS
C      OF (DELX/NSR) TO DELX AND INCREMENTS OF DELX THEREAFTER.
C
C      X      =   DISTANCE ALONG SHOCK
C      ZS     =   DISTANCE ALONG SHOCK AXIS OF SYMMETRY
C      RS     =   RADIUS OF SHOCK
C      RC     =   RADIUS OF CURVATURE OF SHOCK
C      COST   =   COS OF ANGLE OF ATTACK
C      SINT   =   SIN OF ANGLE OF ATTACK
C
C      IMPLICIT REAL*8(A-H,O-Z)
C      COMMON /A9/ EVIS(25),XPST(100)
C      COMMON /A12/ SINTM(1500),COSTM(1500),RSM(1500),RCM(1500),X1(1500)
C      *      ,ZSM(1500)
C      COMMON /A13/ SP,TS,DELX,ZSTERM,IZTERM,NSR,MW
C      DIMENSION ZDUM(501),RSDUM(501),RCDUM(501),XC(501),CODUM(501),
C      *      SIDUM(501)
C      Rs/L = SQRT(2*R*Zs/L - Bs*(Zs/L)**2)      WHERE: L = 1.0 cm
C
C      ITK = 0
C      EL = 1.0
C      R = 230.0
C      BS = -4.0
C      DO 5 I = 2,501
5      ZDUM(I) = (I-1)*ZSTERM/500
C      ZDUM(1) = 0.0
C      RSDUM(1) = 0.0
C      XC(1) = 0.0
C      CODUM(1) = 0.0
C      SIDUM(1) = 1.0
C      RCDUM(1) = 230.0
C
C      DO 10 I = 2,501
C      ZSND = ZDUM(I)/EL
C      RSDUM(I) = DSQRT(2.*R*ZSND - BS*ZSND**2)
C      C = 1.0/(R - BS*ZSND)**2
C      C1 = DSQRT(C*RSDUM(I)**2 + 1.0)
C      XC(I) = RSDUM(I)/2.*C1 + 1./(2.*DSQRT(C))*DLOG(RSDUM(I)*DSQRT(C)
C      *      + C1)
C      D1 = (R - BS*ZSND)/RSDUM(I)
C      D2 = -((R - BS*ZSND)**2/RSDUM(I)**3) - BS/RSDUM(I)
C      THETA = ATAN(D1)
C      SIDUM(I) = DSIN(THETA)
C      CODUM(I) = DCOS(THETA)
C      RCDUM(I) = (1.0 + D1**2)**1.5/DABS(D2)
C
C      DIMENSIONALIZED QUANTITIES
C
C      RCDUM(I) = EL*RCDUM(I)
C      XC(I) = EL*XC(I)
C      RSDUM(I) = EL*RSDUM(I)
C
C      10 CONTINUE
C
C      WRITE(8,25)
C      25 FORMAT(5X,'X',11X,'ZS',10X,'RS',10X,'RC',9X,'COST',8X,'SINT'/)
C      NOX = 0
C      M = 1
C      NZS = 501
C
C      INTERPOLATE SHOCK VALUES TO X SHOCK COORDINATE
C
C      30 NOX = NOX + 1
C      IF (NOX.LT.(NSR+2)) X = X + DELX/NSR
C      IF (NOX.EQ.1) X = 0.0
C      IF (NOX.GT.(NSR+1)) X = X + DELX
C      CALL FTLUP (X,ZS,M,NZS,XC,ZDUM)
C      CALL FTLUP (X,RS,M,NZS,XC,RSDUM)
C      CALL FTLUP (X,RC,M,NZS,XC,RCDUM)
C      CALL FTLUP (X,COST,M,NZS,XC,CODUM)
C      CALL FTLUP (X,SINT,M,NZS,XC,SIDUM)
C      WRITE(8,35) X,ZS,RS,RC,COST,SINT,NOX
C      35 FORMAT(1X,6(E10.4,2X),I4)
C      SINTM(NOX) = SINT
C      COSTM(NOX) = COST
C      RSM(NOX) = RS
C      RCM(NOX) = RC
C      X1(NOX) = X
C      ZSM(NOX) = ZS
```

```

DO 36 IT = 2,NXPSTM
IF (DABS(XPST(IT) - X).GT.1.OE-06) GO TO 36
ITK = ITK + 1
36 CONTINUE
IF (ZS.GE.O.O) GO TO 40
WRITE(15,37) ZS
37 FORMAT(1X,'ZS SHOULD BE GREATER THAN O.O : ZS = ',E10.5,
* /,1X,'STOP 13'/)
STOP
C
40 IF (ZS.LT.ZSTERM) GO TO 30
IZTERM = NOX
RETURN
END
C
C-----
C
SUBROUTINE CALITH (N,CIMAX,PHMAX)
C
C   IN THE CALITH VERSION OF CALINT, THE VARIABLE IN VAR(2) AND
C   CUVAR(2) MAY BE + OR -. VALUES OF OTHER DEPENDENT VARIABLES
C   ARE EXPECTED TO BE POSITIVE.
C
C   10-69
C
IMPLICIT REAL*8(A-H,O-Z)
COMMON /A1/ TE1,TE2,M,IX,IMAX,IPSI,MODEL
COMMON /A2/ P(500),DPDX(500),VARI(500)
COMMON /A3/ EVIINF(25),THETAI(25),MUI(25),FI(25),DELHI(25),
* CIINF(25),LI(25)
COMMON /A4/ R,MUINF,DELTA,LAMSQ,OMEGSQ,OMEGA,MU,T,U,TINF
COMMON /A5/ EPSIIL(20,25),GIL(20,25),CID(200,50)
COMMON /A6/ VAR(52),CUVAR(52),DER(51)
COMMON /A7/ MJ(50),EJ(50),AJ(50),BJ(50),DIRECT(50)
COMMON /A8/ TVI(25),DGENI(25),BI(25),NI(25)
COMMON /A10/ NUIJ(26,50),AIJ(25,50),NUIPJ(25,50)
COMMON /A11/ SIGIK(25,25),ALPIK(25,25),BETAIK(25,25)
COMMON /A17/ ELB,SPEC,CJ,TPREV,HPREV,HCHECK,TCHECK
COMMON /A19/ ELE1(51),ELE2(51),NERR
REAL*8 MUI,MU,MUINF,LAMSQ
DIMENSION F1(51),F2(51),F3(51),CAPF1(51),CAPF2(51),CAPF3(51)
* ,P1(51),PH(51),DELT(51),Y3(51),Y4(51),F4(51),Y2(51)
C
C IF CJ OR N IS EQUAL TO ZERO : NERR = 1
C IF ELE1 IS LESS THAN OR EQUAL TO ELE2 : NERR = 2
C
C   TEST INPUT
C
FN = N
TEST = CJ*FN
IF (TEST) 998,997,998
997 NERR = 1
RETURN
998 DO 999 I = 1,N
IF ((ELE1(I) - ELE2(I)).LE.O.O) THEN
NERR = 2
RETURN
ENDIF
999 CONTINUE
1000 IF (SPEC) 5,1,5
C
C   SECTION FOR INITIALIZATION COMPUTATION OF DERIVATIVES
C
1 SPEC = CJ
ICONT = 1
2 N1 = N + 1
DO 3 I = 1,N1
3 CUVAR(I) = VAR(I)
CALL BASIC
C
C   RETURN WITH DERIVATIVES IN DER
C
DO 4 I = 1,N
4 F1(I) = DER(I)
RETURN
C
C   COMPUTE Y2,X2
C
5 CUVAR(1) = VAR(1) + CJ/2.O
DO 6 I = 1,N
I1 = I + 1
Y2(I) = VAR(I1) + CJ/2.O*F1(I)
IF (I.EQ.1) GO TO 6
IF (Y2(I)) 65,6,6
6 CUVAR(I1) = Y2(I)
GO TO 66
65 SPEC = CJ
CJ = CJ/2.O
IF (CJ.LE.1.OE-25) THEN
WRITE(15,25) CJ

```

```

25  FORMAT(1X,'MIN. INTEGRATION STEP OF ',E12.4,' WAS REACHED'//)
    STOP
    ENDIF
    GO TO 5
C                                     CALL BASIC TO EVALUATE F2
66  CALL BASIC
C                                     RETURN
C
    DO 7 I = 1,N
    I1 = I + 1
    F2(I) = DER(I)
C                                     COMPUTE Y3
    Y3(I) = VAR(I1) + CJ/2.0*F2(I)
    IF (I.EQ.1) GO TO 7
    IF (Y3(I)) 65,7,7
    7  CUVAR(I1) = Y3(I)
C                                     CALL BASIC TO EVALUATE F3
C
    CALL BASIC
C                                     RETURN
C
    DO 10 I = 1,N
    F3(I) = DER(I)
C                                     COMPUTE P,PH AND CAP F TERMS
    IF (Y3(I) - Y2(I)) 9,8,9
    8  P1(I) = 0.0
    GO TO 91
    9  P1(I) = -((F3(I) - F2(I))/(Y3(I) - Y2(I)))
    91  PH(I) = P1(I)*CJ
    IF (PH(I)) 83,83,103
    83  PH(I) = 0.0
    P1(I) = 0.0
    GO TO 84
    103 Z1 = DABS(Y3(I) - Y2(I))/((DABS(Y3(I)) + DABS(Y2(I)))/2.0)
    IF (Z1 - 0.5E-04) 83,83,84
    84  IF (PH(I) - 0.1) 85,85,95
    85  CAPF1(I) = 1.0 - PH(I)/2.0 + PH(I)**2/6.0 - PH(I)**3/24.0
    CAPF2(I) = 0.5 - PH(I)/6.0 + PH(I)**2/24.0 - PH(I)**3/120.0
    CAPF3(I) = 1./6. - PH(I)/24. + PH(I)**2/120. - PH(I)**3/720.
    GO TO 10
    95  CAPF1(I) = (DEXP(-PH(I)) - 1.0)/(- PH(I))
    CAPF2(I) = (CAPF1(I) - 1.0)/(- PH(I))
    CAPF3(I) = (CAPF2(I) - 0.5)/(- PH(I))
    10  CONTINUE
C                                     IS PH BETWEEN ELE2 AND ELE1
    IF (ICONT - 1) 101,101,102
    102  ICONT = ICONT - 1
    SPEC = CJ
    GO TO 17
    101  DO 11 I = 1,N
    IF (DABS(PH(I)) - ELE1(I)) 11,11,13
    11  CONTINUE
    SPEC = CJ
    GO TO 15
C                                     HALVE INTERVAL AND DOUBLE PH RANGE
    13  DO 96 I = 1,N
    ELE1(I) = ELE1(I)*2.0
    IF (ELE1(I) - PHMAX) 94,94,955
    94  ELE2(I) = ELE2(I)*2.0
    GO TO 96
    955  ELE1(I) = ELE1(I)/2.0
    96  CONTINUE
    SPEC = CJ
    CJ = CJ/2.0
    ICONT = 3
    GO TO 5
C                                     RETURN TO RECOMPUTE INTERVAL
    15  DO 16 I = 1,N
    IF (DABS(PH(I)) - ELE2(I)) 16,17,17
    16  CONTINUE
C                                     DOUBLE INTERVAL
    CJ = 2.0*CJ
    IF (CJ - CIMAX) 17,17,165
    165  CJ = CIMAX
C                                     COMPUTE Y4,X4
    17  DO 18 I = 1,N
    I1 = I + 1
    CUVAR(I1) = VAR(I1) + SPEC*(F3(I)*(2.0*CAPF2(I)) + F1(I)*
    * (CAPF1(I) - 2.0*CAPF2(I)) + F2(I)*PH(I)*CAPF2(I))
    IF (I.EQ.1) GO TO 18
    IF (CUVAR(I1)) 175,18,18
    175  CJ = SPEC
    CJ = CJ/2.0

```

```

GO TO 5
18 Y4(I) = CUVAR(I1)
   CUVAR(1) = VAR(1) + SPEC
C
C      CALL BASIC                                CALL BASIC TO EVALUATE F4
C
C      RETURN
C
DO 20 I = 1,N
I1 = I + 1
F4(I) = DER(I)
C
C      COMPUTE DELTA Y
C
DELTY(I) = SPEC*(F1(I)*CAPF1(I)+(-3.0*(F1(I)+P1(I)*VAR(I1))+2.0*
* (F2(I)+P1(I)*Y2(I)) + 2.0*(F3(I)+P1(I)*Y3(I)) - F4(I) - P1(I)*
* Y4(I))*CAPF2(I) + 4.0*((F1(I)+P1(I)*VAR(I1)) - (F2(I)+P1(I)*
* Y2(I)) - (F3(I)+P1(I)*Y3(I)) + (F4(I)+P1(I)*Y4(I)))*CAPF3(I))
C
C      COMPUTE Y + DELTA Y
C
20 CUVAR(I1) = VAR(I1) + DELTY(I)
C      CALL CHECK FOR DECISION TO ACCEPT OR RECOMPUTE INTERVAL
   CALL CHECK
   IF (ELB) 21,21,23
C
C      UPDATE Y VALUES
C
21 N1 = N + 1
   DO 22 I = 2,N1
   I1 = I - 1
22 VAR(I) = VAR(I) + DELTY(I1)
   VAR(1) = VAR(1) + SPEC
C
C      RETURN TO COMPUTE DERIVATIVES AT Y + DELTA Y
   GO TO 2
C
C      RETURN TO RECOMPUTE INTERVAL
23 GO TO 5
   END
C
-----
C
SUBROUTINE FTLUP (X,Y,M,N,VARI,VARD)
C
C      THIS SUBROUTINE IS A MODIFICATION OF LIBRARY INTERPOLATION
C      SUBROUTINE FTLUP   REVISED 7-7-69
C
C      IMPLICIT REAL*8(A-H,O-Z)
C      DIMENSION VARI(5),VARD(5),V(3),YY(2),II(100)
C      INITIALIZE ALL INTERVAL POINTERS TO -1.0 FOR MONOTONICITY CHECK
DO 4 J = 1,100
4 II(J) = -1
MA = IABS(M)
C      ASSIGN INTERVAL POINTER FOR GIVEN VARI TABLE ; THE SAME POINTER
C      WILL BE USED ON A GIVEN VARI TABLE EVERY TIME
LOCF = VARI(1)
LI = MOD(LOCF,100) + 1
I = II(LI)
IF (I.GE.0) GO TO 10
IF (N.LT.2) GO TO 10
C
C      MONOTONICITY CHECK
C
IF (VARI(2) - VARI(1)) 1,1,3
C      ERROR IN MONOTONICITY
2 K = LOCF
WRITE(15,102) J,K
102 FORMAT(1X,'TABLE BELOW OUT OF ORDER FOR FTLUP AT POSITION ',
* 15,/,1X,'X TABLE IS STORED IN LOCATION',15//)
DO 103 J = 1,N
103 WRITE(15,*) VARI(J),VARD(J)
STOP
C
C      MONOTONIC DECREASING
1 DO 5 J = 2,N
IF (VARI(J) - VARI(J-1)) 5,2,2
5 CONTINUE
GO TO 10
C
C      MONOTONIC INCREASING
3 DO 6 J = 2,N
IF (VARI(J) - VARI(J-1)) 2,2,6
6 CONTINUE
C
C      INTERPOLATION
10 IF (I.LE.0) I = 1
IF (I.GE.N) I = N - 1
IF (N.LE.1) GO TO 8
IF (MA.NE.0) GO TO 99
C
C      ZERO ORDER
8 Y = VARD(1)
GO TO 800

```

```

C                                     LOCATE I INTERVAL (X(I).LE.X.LT.X(I+1))
99 IF ((VARI(I) - X)*(VARI(I+1) - X)) 61,61,40
C                                     IN GIVES DIRECTION FOR SEARCH INTERVALS
40 SIGN1 = 1.0
   IN = SIGN(SIGN1,(VARI(I+1) - VARI(I))*(X - VARI(I)))
C                                     IF X OUTSIDE ENDPOINTS, EXTRAPOLATE FROM END INTERVAL
41 IF ((I+IN).LE.O) GO TO 61
   IF ((I+IN).GE.N) GO TO 61
   I = I + IN
   IF ((VARI(I) - X)*(VARI(I+1) - X)) 61,61,41
61 IF (MA.EQ.2) GO TO 200
C                                     FIRST ORDER
   Y = (VARD(I)*(VARI(I+1) - X) - VARD(I+1)*(VARI(I) - X))/
   *   (VARI(I+1) - VARI(I))
   GO TO 800
C                                     SECOND ORDER
200 IF (N.EQ.2) GO TO 2
   IF (I.EQ.(N-1)) GO TO 209
   IF (I.EQ.1) GO TO 201
C                                     PICK THIRD POINT
   SK = VARI(I+1) - VARI(I)
   IF ((SK*(X - VARI(I-1))).LT.(SK*(VARI(I+2) - X))) GO TO 209
201 L = I
   GO TO 702
209 L = I - 1
702 V(1) = VARI(L) - X
   V(2) = VARI(L+1) - X
   V(3) = VARI(L+2) - X
   YY(1) = (VARD(L)*V(2)-VARD(L+1)*V(1))/(VARI(L+1)-VARI(L))
   YY(2) = (VARD(L+1)*V(3)-VARD(L+2)*V(2))/(VARI(L+2)-VARI(L+1))
   Y = (YY(1)*V(3) - YY(2)*V(1))/(VARI(L+2) - VARI(L))
800 II(LI) = I
   RETURN
   END

```

```

C-----
C
C SUBROUTINE FOFT23 (MAXI,TOL1,TOL2,ICODE,TSG,IQ1,IQ2,IQ3)
C
C CALLED BY MAIN TO EVALUATE TS BY NEWTON ITERATION TECHNIQUE
C
C IMPLICIT REAL*8(A-H,O-Z)
COMMON /A1/ TE1,TE2,M,IX,IMAX,IPSI,MODEL
COMMON /A2/ P(500),DPDX(500),VARI(500)
COMMON /A3/ EVIINF(25),THETAI(25),MUI(25),FI(25),DELHI(25),
* CIINF(25),LI(25)
COMMON /A4/ R,MUINF,DELTA,LAMSQ,OMEGSQ,OMEGA,MU,T,U,TINF
COMMON /A5/ EPSIIL(20,25),GIL(20,25),CID(200,50)
COMMON /A6/ VAR(52),CUVAR(52),DER(51)
DIMENSION SUMGM(25),EISM(25),DEISM(25)
REAL*8 MUI,LAMSQ,MUINF,MU
C
W = 1.0
DO 1 I = 1,IMAX
1 CID(IX,I) = CIINF(I)
ITER = 0
2 ITER = ITER + 1
IF (ITER.EQ.1) THEN
   ES = DELTA
   PS = DSQRT(OMEGSQ - 2.*LAMSQ*(DELTA - ES))
ENDIF
IF (IQ1.EQ.1) THEN
   TVIB = TINF
ELSE
   TVIB = TSG
ENDIF
IF (IQ2.EQ.1) THEN
   TEL = TINF
ELSE
   TEL = TSG
ENDIF
C
ES = 0.0
DES = 0.0
DO 25 I = 1,IMAX
IF (M.EQ.1) EVIS = EVIINF(I)
IF (M.EQ.1) GO TO 5
TEM = DEXP(THETAI(I)/TVIB)
EVIS = (R*THETAI(I))/(MUI(I)*(TEM - 1.0))*FI(I)
DEVIS = (R*FI(I)*THETAI(I)**2)/(MUI(I)*TVIB**2)*(TEM/(TEM-1.0)**2)

```

```

5 IF (I.EQ.1) QVN = 1.0/(1.0 - DEXP(-THETAI(I)/TVIB))
  IF (I.EQ.2) QVO = 1.0/(1.0 - DEXP(-THETAI(I)/TVIB))
  SUMGM(I) = 0.0
  SUMGE = 0.0
  DSUMGE = 0.0
  LII = LI(I)

C
  DO 10 L = 1,LII
    TEM1 = DEXP(-EPSIIL(L,I)/TEL)
    SUMGM(I) = SUMGM(I) + TEM1*GIL(L,I)
    SUMGE = SUMGE + TEM1*GIL(L,I)*EPSIIL(L,I)
10  DSUMGE = DSUMGE + TEM1*GIL(L,I)*EPSIIL(L,I)**2
    QROTN = TSG/5.8
    QROTD = TSG/4.2
    EEIS = (R/MUI(I))*(SUMGE/SUMGM(I))
    DEEIS = R/MUI(I)*(((SUMGM(I)*DSUMGE/TEL**2) - (SUMGE/TEL)**2)
      * /SUMGM(I)**2)
    * EISM(I) = (1.5*R*TSG)/MUI(I) + (FI(I)*R*TSG)/MUI(I) + EVIS + EEIS
    * + DELHI(I)/MUI(I)
    DEISM(I) = 1.5*R/MUI(I) + FI(I)*R/MUI(I) + DEVIS + DEEIS
25  CONTINUE
C
  IF (IQ3.EQ.2) THEN
    TD = 59500.0
    F1 = CIINF(2)*MUINF/MUI(2)
    C1A = TD/TSG
    C1A = DEXP(C1A/2.0)
    C1B = 8.8568195*(1. - CIINF(1))*C1A*(1./(TSG**2.5))
    C1 = C1A*2.0*MUINF*PS*C1B*QVO*QROTD*(SUMGM(2)/(SUMGM(4)**2))/R
    BETAO = ((F1-1.) + DSQRT((1.-F1)**2 + 4.*(F1+C1)))/(2.*(F1+C1))
    BETAN = 0.0
  ENDIF
C
  IF (IQ3.EQ.3) THEN
    BETAN = .1
    BETAO = 1.0
    TDN = 113500.0
    TDO = 59500.0
    F1A = 8.8568195*(1. - CIINF(1))*DEXP(TDO/TSG)*(1./(TSG**2.5))
    F2A = 12.36681*(1. - CIINF(2))*DEXP(TDN/TSG)*(1./(TSG**2.5))
    F1 = 2.0*MUINF*PS*F1A*QVO*QROTD*(SUMGM(2)/(SUMGM(4)**2))/R
    F2 = 2.0*MUINF*PS*F2A*QVN*QROTN*(SUMGM(1)/(SUMGM(3)**2))/R
30  A = ((1. - BETAO**2) + BETAN*(1. - BETAO))/(BETAO**2) - F1
    B = ((1. - BETAN**2) + BETAO*(1. - BETAN))/(BETAN**2) - F2
    ABO = (BETAN*BETAO - 2.*(1. + BETAN))/(BETAO**3)
    ABN = (1. - BETAO)/(BETAO**2)
    BBO = (1. - BETAN)/(BETAN**2)
    BBN = (BETAN*BETAO - 2.*(1. + BETAO))/(BETAN**3)
    DEN = ABO*BBN - BBO*ABN
    DBO = (-A*BBN + B*ABN)/DEN
    DBN = (-B*ABO + A*BBO)/DEN
    BETAN = BETAN + DBN
    BETAO = BETAO + DBO
    IF ((DABS(DBO).GT..00001).OR.(DABS(DBN).GT..00001)) GO TO 30
  ENDIF
C
  DISSOCIATING CONCENTRATIONS ACROSS THE SHOCK
  CID(IX,2) = (1.0 - BETAO)*CIINF(2)
  CID(IX,4) = BETAO*CIINF(2)
  CID(IX,1) = (1.0 - BETAN)*CIINF(1)
  CID(IX,3) = BETAN*CIINF(1)
C
  DO 35 I = 1,IMAX
    ES = ES + EISM(I)*CID(IX,I)
35  DES = DES + DEISM(I)*CID(IX,I)
    FAC = DSQRT(OMEGSQ - 2.0*LAMSQ*(DELTA - ES))
    FIN = (2.0*MUINF*(DELTA - ES)*FAC)/(R*(OMEGA + FAC))
    FTS = TSG - FIN
    FAC1 = R*(OMEGA+FAC)*(MUINF*(DELTA-ES)*(2.0/FAC)*LAMSQ*DES -
      * 2.0*FAC*MUINF*DES) - 2.0*MUINF*(DELTA-ES)*R*LAMSQ*DES
    FAC2 = R**2*(OMEGA + FAC)**2
    DFTS = 1.0 - (FAC1/FAC2)
    PS = FAC
    IF (DFTS.EQ.0.0) THEN
      ICODE = 2
      RETURN
    ENDIF
    TSG1 = TSG - W*(FTS/DFTS)
    DELTS = DABS(TSG1 - TSG)
    TSG = TSG1
    IF (ITER.GE.MAXI) THEN

```



```

        ICODE = 1
        RETURN
    ENDIF
    IF (DELTS.GT.TOL1) THEN
        ITER1 = ITER
        GO TO 2
    ENDIF
    ITER2 = ITER1 + 10
    IF (DELTS.LE.TOL2.OR.ITER.GE.ITER2) RETURN
    GO TO 2
END

```

C
C-----
C

```

SUBROUTINE OLRAD (IQ4,IQ5,IQ6,IQ8,IQ9,IQ10)
IMPLICIT REAL*8(A-H,O-Z)
COMMON /A1/ TE1,TE2,M,IX,IMAX,IPSI,MODEL
COMMON /A3/ EVIINF(25),THETAI(25),MUI(25),FI(25),DELHI(25),
* CIINF(25),LI(25)
COMMON /A5/ EPSIIL(20,25),GIL(20,25),CID(200,50)
COMMON /A9/ EVIS(25),XPST(100)
COMMON /RAD/ RCON(100,10,40),RRHO(100,40),TEMP(100,40)
* ,TSTAG,Y(100,40),NXCON,NPTS(100)
COMMON /RADC/ YBDY(100),B(100,8,40),DPTL(100,8,40)
COMMON /RADO/ TEMPO(100,40),ALPN(100,40),ALPO(100,40)
COMMON /RADDL/ BETN(100,40),BETO(100,40)
REAL*8 KAPPA(100,8,40),KAP(100,4,40),KA(100,4,40),MUI
REAL*8 K11,K22,N,NN
WRITE(8,1)
1 FORMAT(///' OLRAD RADIATION MODEL')
IF(IQ8.EQ.0) THEN
    IF(IQ6.EQ.0) WRITE(8,2)
    IF(IQ6.EQ.1) WRITE(8,3)
ENDIF
2 FORMAT('/ BETA=BETA(Te)')
3 FORMAT('/ BETA=BETA(Tt)')

```

C

NUMBER DENSITIES

```

DO 12 K=3,NXCON
DO 12 J=1,NPTS(K)
DO 10 II=1,IMAX
XXX=RRHO(K,J)*6.023D+23
XXX=XXX*RCON(K,II,J)/MUI(II)
10 RCON(K,II,J)=XXX
12 CONTINUE
    IF(IQ10.EQ.0)GO TO 15
    DO 14 K=3,NXCON
    NUP=NPTS(K)-1
    DO 14 J=1,NUP
    FNA=RCON(K,3,J)
    FNI=RCON(K,9,J)
    FNE=RCON(K,7,J)
    FNM=RCON(K,1,J)
    EI=2.3322D-11
    FAV=6.023D23
    SA=1.52D-15
    TE=TEMP(K,J)
16 SM=0.5355D-19*TE+0.696D-15
    SI=1.53614D08*(TE**3)/FNE
    SI=DSQRT(SI)
    SI=(4.38384D-06/(TE**2))*DLOG(SI)
    FKF=1.1D32*(TE**(-3.14D0))*DEXP(-1.69D05/TE)
    FKB=2.2D20*(TE**(-4.5D00))*1.D20
    WEA=FKF*FNA-FKB*FNE*(FNI/FAV)
    SX=FNA*SA+FNI*SI+FNM/2.0D0*SM
    W1=WEA
    T1=TEMPO(K,J)-1.23357D-10/SX/DSQRT(TE)*(WEA*EI+W1*3.45D-16*TE)
    IF(T1.LT.0.0D00)T1=0.0D0
    TT=DABS(TE-T1)
    TE=TE+(T1-TE)*0.125D00
    IF(TT.GT.1.0D00)GO TO 16
    IF(TE.GT.TEMPO(K,J))TE=TEMPO(K,J)
    TEMP(K,J)=TE
14 CONTINUE
15 CONTINUE

```

C

NONEQUILIBRIUM RADIATION CORRECTION

```

IF (IQ5.EQ.0) THEN
DO 122 K=3,NXCON
DO 122 J=1,NPTS(K)
IF(J.EQ.NPTS(K)) THEN
    ALPN(K,J)=0.D+00

```

```

      BETN(K,J)=O.D+OO
      GO TO 122
    ENDIF
    Q1=O.D+O
    Q2=O.D+O
    Q3=O.D+O
    DO 112 III=1,LI(3)
112   Q1=Q1+GIL(III,3)*DEXP(-EPSIIL(III,3)/TEMP(K,J))
    DO 113 III=1,LI(9)
113   Q2=Q2+GIL(III,9)*DEXP(-EPSIIL(III,9)/TEMP(K,J))
    DO 130 III=1,LI(1)
130   Q3=Q3+GIL(III,1)*DEXP(-EPSIIL(III,1)/TEMP(K,J))
      RH01=RCON(K,3,J)*1.401D+O1/6.023D+23
      RH02=RCON(K,9,J)*1.401D+O1/6.023D+23
      RH03=RCON(K,1,J)*2.802D+O1/6.023D+23
      RH04=RH01+RH02+RH03
      QVIB=1.D+O/(1.D+O-DEXP(-3.39D+O3/TEMP(K,J)))
      QROT=TEMP(K,J)/5.8D+O
      C1=RH04*2.4701D+O1*QVIB*QROT*Q3*DEXP(1.135D+O5/TEMP(K,J))
      C1=C1/(Q1**2*TEMP(K,J)**1.5)
      C2=RH04*8.909D+O6*Q1*DEXP(1.69D+O5/TEMP(K,J))
      C2=C2/(Q2*TEMP(K,J)**1.5)
      ALPHA=O.5D+O
      BETA=O.5D+O
132   F=BETA**2*(1.D+O-ALPHA)**2*C1+BETA-1.D+O
      FA=-2.D+O*BETA**2*(1.D+O-ALPHA)*C1
      FB=2.D+O*BETA*(1.D+O-ALPHA)**2*C1+1.D+O
      G=ALPHA**2*BETA*C2+ALPHA-1.D+O
      GA=2.D+O*ALPHA*BETA*C2+1.D+O
      GB=ALPHA**2*C2
      DENOM=FA*GB-FB*GA
      DA=(-F*GB+G*FB)/DENOM
      DB=(-G*FA+F*GA)/DENOM
      ALPHA=ALPHA+DA
      BETA=BETA+DB
      IF((DABS(DA).GT.1.D-O5).OR.(DABS(DB).GT.1.D-5)) GO TO 132
      ALPH=RCON(K,9,J)/(RCON(K,9,J)+RCON(K,3,J))
      BET=(RCON(K,9,J)+RCON(K,3,J))/(2.D+O*RCON(K,1,J)+
*      RCON(K,9,J)+RCON(K,3,J))
      IF(IQ6.EQ.1) GO TO 1888
      BETN(K,J)=(BET**2/(1.D+O-BET))*((1.D+O-BETA)/(BETA**2))
*      *((1.D+O-ALPH)**2/(1.D+O-ALPHA)**2)
1888  ALPN(K,J)=BET*(ALPH**2/(1.D+O-ALPH))
*      *((1.D+O-ALPHA)/(BETA*ALPHA**2))
122  CONTINUE
      DO 142 K=3,NXCON
      DO 142 J=1,NPTS(K)
      IF(J.EQ.NPTS(K)) THEN
        ALPO(K,J)=O.D+OO
        BETO(K,J)=O.D+OO
        GO TO 142
      ENDIF
      Q1=O.D+O
      Q2=O.D+O
      Q3=O.D+O
      DO 152 III=1,LI(4)
152   Q1=Q1+GIL(III,4)*EXP(-EPSIIL(III,4)/TEMP(K,J))
      DO 153 III=1,LI(10)
153   Q2=Q2+GIL(III,10)*DEXP(-EPSIIL(III,10)/TEMP(K,J))
      DO 160 III=1,LI(2)
160   Q3=Q3+GIL(III,2)*DEXP(-EPSIIL(III,2)/TEMP(K,J))
      RH01=RCON(K,4,J)*1.600D+O1/6.023D+23
      RH02=RCON(K,10,J)*1.600D+O1/6.023D+23
      RH03=RCON(K,2,J)*3.200D+O1/6.023D+23
      RH04=RH01+RH02+RH03
      QVIB=1.D+O/(1.D+O-DEXP(-2.27D+O3/TEMP(K,J)))
      QROT=TEMP(K,J)/4.2D+O
      C1=RH04*1.7725D+O1*QVIB*QROT*Q3*DEXP(5.950D+O4/TEMP(K,J))
      C1=C1/(Q1**2*TEMP(K,J)**1.5)
      C2=RH04*7.801D+O6*Q1*DEXP(1.58D+O5/TEMP(K,J))
      C2=C2/(Q2*TEMP(K,J)**1.5)
      ALPHA=O.5D+O
      BETA=O.5D+O
162   F=BETA**2*(1.D+O-ALPHA)**2*C1+BETA-1.D+O
      FA=-2.D+O*BETA**2*(1.D+O-ALPHA)*C1
      FB=2.D+O*BETA*(1.D+O-ALPHA)**2*C1+1.D+O
      G=ALPHA**2*BETA*C2+ALPHA-1.D+O
      GA=2.D+O*ALPHA*BETA*C2+1.D+O
      GB=ALPHA**2*C2
      DENOM=FA*GB-FB*GA

```

```

DA=(-F*GB+G*FB)/DENOM
DB=(-G*FA+F*GA)/DENOM
ALPHA=ALPHA+DA
BETA=BETA+DB
IF((DABS(DA).GT.1.D-05).OR.(DABS(DB).GT.1.D-5)) GO TO 162
ALPH=RCON(K,10,J)/(RCON(K,10,J)+RCON(K,4,J))
BET=(RCON(K,10,J)+RCON(K,4,J))/(2.D+O*RCON(K,2,J)+
* RCON(K,10,J)+RCON(K,4,J))
IF(IQ6.EQ.1) GO TO 1999
BETO(K,J)=(BET**2/((1.D+O-BET))*((1.D+O-BETA)/(BETA**2))
* ((1.D+O-ALPH)**2/((1.D+O-ALPHA)**2))
1999 ALPO(K,J)=BET*(ALPH**2/((1.D+O-ALPH))
* ((1.D+O-ALPHA)/(BETA*ALPHA**2))
142 CONTINUE
C BETA=BETA(Tt)
IF(IQ6.EQ.1) THEN
DO 1122 K=3,NXCON
DO 1122 J=1,NPTS(K)
IF(J.EQ.NPTS(K)) THEN
BETN(K,J)=O.D+OO
GO TO 1122
ENDIF
Q1=O.D+O
Q2=O.D+O
Q3=O.D+O
DO 1112 III=1,LI(3)
1112 Q1=Q1+GIL(III,3)*DEXP(-EPSIIL(III,3)/TEMPO(K,J))
DO 1113 III=1,LI(9)
1113 Q2=Q2+GIL(III,9)*DEXP(-EPSIIL(III,9)/TEMPO(K,J))
DO 1130 III=1,LI(1)
1130 Q3=Q3+GIL(III,1)*DEXP(-EPSIIL(III,1)/TEMPO(K,J))
RHO1=RCON(K,3,J)*1.401D+01/6.O23D+23
RHO2=RCON(K,9,J)*1.401D+01/6.O23D+23
RHO3=RCON(K,1,J)*2.802D+01/6.O23D+23
RHO4=RHO1+RHO2+RHO3
QVIB=1.D+O/((1.D+O-DEXP(-3.39D+03/TEMPO(K,J)))
QROT=TEMPO(K,J)/5.8D+O
C1=RHO4*2.4701D+01*QVIB*QROT*Q3*DEXP(1.135D+05/TEMPO(K,J))
C1=C1/(Q1**2*TEMPO(K,J)**1.5)
C2=RHO4*8.909D+06*Q1*DEXP(1.69D+05/TEMPO(K,J))
C2=C2/(Q2*TEMPO(K,J)**1.5)
ALPHA=O.5D+O
BETA=O.5D+O
1132 F=BETA**2*(1.D+O-ALPHA)**2*C1+BETA-1.D+O
FA=-2.D+O*BETA**2*(1.D+O-ALPHA)*C1
FB=2.D+O*BETA*(1.D+O-ALPHA)**2*C1+1.D+O
G=ALPHA**2*BETA*C2+ALPHA-1.D+O
GA=2.D+O*ALPHA*BETA*C2+1.D+O
GB=ALPHA**2*C2
DENOM=FA*GB-FB*GA
DA=(-F*GB+G*FB)/DENOM
DB=(-G*FA+F*GA)/DENOM
ALPHA=ALPHA+DA
BETA=BETA+DB
IF((DABS(DA).GT.1.D-05).OR.(DABS(DB).GT.1.D-5)) GO TO 1132
BET=(RCON(K,9,J)+RCON(K,3,J))/(2.D+O*RCON(K,1,J)+
* RCON(K,9,J)+RCON(K,3,J))
ALPH=RCON(K,9,J)/(RCON(K,9,J)+RCON(K,3,J))
BETN(K,J)=(BET**2/((1.D+O-BET))*((1.D+O-BETA)/(BETA**2))
* ((1.D+O-ALPH)**2/((1.D+O-ALPHA)**2))
1122 CONTINUE
DO 1142 K=3,NXCON
DO 1142 J=1,NPTS(K)
IF(J.EQ.NPTS(K)) THEN
BETO(K,J)=O.D+OO
GO TO 1142
ENDIF
Q1=O.D+O
Q2=O.D+O
Q3=O.D+O
DO 1152 III=1,LI(4)
1152 Q1=Q1+GIL(III,4)*EXP(-EPSIIL(III,4)/TEMPO(K,J))
DO 1153 III=1,LI(10)
1153 Q2=Q2+GIL(III,10)*DEXP(-EPSIIL(III,10)/TEMPO(K,J))
DO 1160 III=1,LI(2)
1160 Q3=Q3+GIL(III,2)*DEXP(-EPSIIL(III,2)/TEMPO(K,J))
RHO1=RCON(K,4,J)*1.600D+01/6.O23D+23
RHO2=RCON(K,10,J)*1.600D+01/6.O23D+23
RHO3=RCON(K,2,J)*3.200D+01/6.O23D+23
RHO4=RHO1+RHO2+RHO3

```

```

QVIB=1.D+O/(1.D+O-DEXP(-2.27D+O3/TEMPO(K,J)))
QROT=TEMPO(K,J)/4.2D+O
C1=RH04*1.7725D+O1*QVIB*QROT*Q3*DEXP(5.950D+O4/TEMPO(K,J))
C1=C1/(Q1**2*TEMPO(K,J)**1.5)
C2=RH04*7.801D+O6*Q1*DEXP(1.58D+O5/TEMPO(K,J))
C2=C2/(Q2*TEMPO(K,J)**1.5)
ALPHA=O.5D+O
BETA=O.5D+O
1162 F=BETA**2*(1.D+O-ALPHA)**2*C1+BETA-1.D+O
FA=-2.D+O*BETA**2*(1.D+O-ALPHA)*C1
FB=2.D+O*BETA*(1.D+O-ALPHA)**2*C1+1.D+O
G=ALPHA**2*BETA*C2+ALPHA-1.D+O
GA=2.D+O*ALPHA*BETA*C2+1.D+O
GB=ALPHA**2*C2
DENOM=FA*GB-FB*GA
DA=(-F*GB+G*FB)/DENOM
DB=(-G*FA+F*GA)/DENOM
ALPHA=ALPHA+DA
BETA=BETA+DB
IF((DABS(DA).GT.1.D-05).OR.(DABS(DB).GT.1.D-5)) GO TO 1162
BET=(RCON(K,10,J)+RCON(K,4,J))/(2.D+O*RCON(K,2,J)+
* RCON(K,10,J)+RCON(K,4,J))
ALPH=RCON(K,10,J)/(RCON(K,10,J)+RCON(K,4,J))
BETO(K,J)=(BET**2/(1.D+O-BET))*((1.D+O-BETA)/(BETA**2))
* ((1.D+O-ALPH)**2/(1.D+O-ALPHA)**2)
1142 CONTINUE
ENDIF
ENDIF
C ABSORPTION COEFFICIENTS
DO 20 K=3,NXCON
DO 20 J=1,NPTS(K)
TT=TEMP(K,J)/168800.O
IF(IQ6.EQ.1) THEN
TM=TEMPO(K,J)/168800.O
ELSE
TM=TT
ENDIF
C IF((IQ6.EQ.1).AND.(TEMPO(K,J).LE.8.D+O3)) GO TO 20
C IF(TEMP(K,J).LE.8.D+O3) GO TO 20
IF(IQ8.EQ.0) THEN
O=BETO(K,J)
N=BETN(K,J)
ELSE
O=1.D+O
N=1.D+O
ENDIF
KAPPA(K,4,J)=5.D-19*O*RCON(K,2,J)+5.D-20*N*RCON(K,1,J)
KAP(K,4,J)=1.7D-17*RCON(K,3,J)*DEXP(-.246/TT)
KA(K,4,J)=O.D+O
KAPPA(K,3,J)=2.D-18*(N*RCON(K,1,J)+O*RCON(K,2,J))+KAPPA(K,4,J)
KAP(K,3,J)=2.1D-17*RCON(K,3,J)*DEXP(-.165/TT)+KAP(K,4,J)
KA(K,3,J)=O.D+O
KAPPA(K,2,J)=5.1D-18*(N*RCON(K,1,J)+O*RCON(K,2,J))+KAPPA(K,3,J)
KAP(K,2,J)=KAP(K,3,J)
KA(K,2,J)=5.1D-18*RCON(K,4,J)
KAPPA(K,1,J)=2.D-17*O*RCON(K,2,J)+4.D-16*N*RCON(K,1,J)
C +KAPPA(K,2,J)
KAP(K,1,J)=1.1D-17*RCON(K,3,J)+KAP(K,2,J)
KA(K,1,J)=KA(K,2,J)
IF(IQ5.EQ.0) THEN
N=ALPN(K,J)
O=ALPO(K,J)
IF(IQ8.EQ.0) THEN
NN=BETN(K,J)
OO=BETO(K,J)
ELSE
NN=1.D+O
OO=1.D+O
ENDIF
KAPPA(K,5,J)=7.7D-17*(NN*RCON(K,1,J)+OO*RCON(K,2,J))
C *DEXP(-.49/TM)+
C 2.6D-17*(RCON(K,3,J)*N+RCON(K,4,J)*O)*DEXP(-.723/TT)
KAPPA(K,6,J)=2.D-18*RCON(K,2,J)*OO+6.OD-18*(RCON(K,3,J)*N+
C RCON(K,4,J)*O)*DEXP(-.379/TT)+KAPPA(K,5,J)
IF(RCON(K,7,J).LE.1.D+OO) THEN
KAPPA(K,7,J)=KAPPA(K,6,J)
GO TO 909
ENDIF
KAPPA(K,7,J)=1.2D03*((RCON(K,3,J)*N+RCON(K,4,J)*O)/RCON(K,7,J))*
C DEXP(-.489/TT)+KAPPA(K,6,J)

```

```

909 KAPPA(K,8,J)=3.2D-17*(RCON(K,3,J)*N+RCON(K,4,J)*O)*DEXP(-.631/TT)
C      +KAPPA(K,5,J)
      ELSE
      KAPPA(K,5,J)=7.7D-17*(RCON(K,1,J)+RCON(K,2,J))*DEXP(-.49/TT)+
C      2.6D-17*(RCON(K,3,J)+RCON(K,4,J))*DEXP(-.723/TT)
      KAPPA(K,6,J)=2.D-18*RCON(K,2,J)+6.0D-18*(RCON(K,3,J)+RCON(K,4,J))
C      *DEXP(-.379/TT)+KAPPA(K,5,J)
      IF(RCON(K,7,J).LE.1.D+00) THEN
      KAPPA(K,7,J)=KAPPA(K,6,J)
      GO TO 910
      ENDIF
      KAPPA(K,7,J)=1.2D03*((RCON(K,3,J)+RCON(K,4,J))/RCON(K,7,J))*
C      DEXP(-.489/TT)+KAPPA(K,6,J)
910 KAPPA(K,8,J)=3.2D-17*(RCON(K,3,J)+RCON(K,4,J))*DEXP(-.631/TT)
C      +KAPPA(K,5,J)
      ENDIF
20 CONTINUE

C      PLANCK FUNCTION
      DO 30 K=3,NXCON
      DO 30 J=1,NPTS(K)
      TT=TEMP(K,J)/168800.0
C      IF(TEMP(K,J).LE.8.D+03) GO TO 30
      BLACK=(TT*168800.)**4*5.6696D-12/3.1415927
      FF=15.D+00/(3.1415927**4)
      X1=1./TT
      B1=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C      DEXP(-2.*X1)*0.5*(.75+1.5*X1+1.5*X1**2+X1**3))
      X1=0.935/TT
      B2=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C      DEXP(-2.*X1)*0.5*(.75+1.5*X1+1.5*X1**2+X1**3))
      X1=0.835/TT
      B3=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C      DEXP(-2.*X1)*0.5*(.75+1.5*X1+1.5*X1**2+X1**3))
      X1=0.754/TT
      B4=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C      DEXP(-2.*X1)*0.5*(.75+1.5*X1+1.5*X1**2+X1**3))
      X1=0.473/TT
      B6=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C      DEXP(-2.*X1)*0.5*(.75+1.5*X1+1.5*X1**2+X1**3))
      X1=0.213/TT
      B8=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C      DEXP(-2.*X1)*0.5*(.75+1.5*X1+1.5*X1**2+X1**3))
      B8=1-B8
      B2=B2-B1
      B3=B3-B2
      B4=B4-B3
      B6=B6-B4
      B5=1-B8-B6-B4-B3-B2-B1
      B(K,1,J)=BLACK*B1
      B(K,2,J)=BLACK*B2
      B(K,3,J)=BLACK*B3
      B(K,4,J)=BLACK*B4
      B(K,6,J)=BLACK*B6
      B(K,7,J)=BLACK*2.4D-21*RCON(K,7,J)*B6*DEXP(.162/TT)
      B(K,8,J)=BLACK*B8
      B(K,5,J)=BLACK*B5
30 CONTINUE

C      TAU'S
      DO 100 K=3,NXCON
      DO 100 I=1,8
      IF(I.GT.4) THEN
      OPTL(K,I,1)=KAPPA(K,I,1)*DABS(Y(K,1)-YBDY(K))
      ELSE
C      OPTL(K,I,1)=(KAPPA(K,I,1)+KAP(K,I,1)+KA(K,I,1))
      *DABS(Y(K,1)-YBDY(K))
      ENDIF
      DO 100 J=2,NPTS(K)
      IF(I.GT.4) THEN
      K11=KAPPA(K,I,J)+KAPPA(K,I,J-1)
      ELSE
C      K11=KAPPA(K,I,J)+KAP(K,I,J)+KAPPA(K,I,J-1)+KAP(K,I,J-1)
      +KA(K,I,J)+KA(K,I,J-1)
      ENDIF
      OPTL(K,I,J)=OPTL(K,I,J-1)+(5.D-1*(K11)*DABS(Y(K,J-1)
C      -Y(K,J)))
100 CONTINUE

C      QRW
      DO 200 K=3,NXCON
      WRITE(8,102) XPST(K)
102 FORMAT(///, ' AT X = ',D11.4)

```

```

WQR=0.0
DO 180 I=1,8
CALL FEI(E1I,E2I,E3I,OPTL(K,I,1))
EOLD=E2I
JIN=1
IF(I.GT.4) THEN
  K11=KAPPA(K,I,1)*B(K,I,1)
ELSE
  IF(IQ5.EQ.1) THEN
    ALPN(K,1)=1.D+0
    ALPO(K,1)=1.D+0
  ENDIF
  K11=KAPPA(K,I,1)*B(K,I,1)+KAP(K,I,1)*B(K,I,1)*ALPN(K,1)+
C   KA(K,I,1)*B(K,I,1)*ALPO(K,1)
ENDIF
SUM=3.1415927*EOLD*K11*DABS(Y(K,1)-YBDY(K))
IF(IQ9.EQ.0) WRITE(8,2221) JIN,I,SUM
DO 150 J=2,NPTS(K)
IF(I.GT.4) THEN
  K11=KAPPA(K,I,J)*B(K,I,J)
  K22=KAPPA(K,I,J-1)*B(K,I,J-1)
ELSE
  IF(IQ5.EQ.1) THEN
    ALPN(K,J)=1.D+0
    ALPO(K,J)=1.D+0
  ENDIF
  K11=KAPPA(K,I,J)*B(K,I,J)+KAP(K,I,J)*B(K,I,J)*ALPN(K,J)+
C   KA(K,I,J)*B(K,I,J)*ALPO(K,J)
  K22=KAPPA(K,I,J-1)*B(K,I,J-1)+
C   KAP(K,I,J-1)*B(K,I,J-1)*ALPN(K,J-1)+
C   KA(K,I,J-1)*B(K,I,J-1)*ALPO(K,J-1)
ENDIF
Z=OPTL(K,I,J)
CALL FEI(E1I,E2I,E3I,Z)
STSUM=3.14159*(K11*E2I+K22*EOLD)*DABS(Y(K,J-1)-Y(K,J))
IF(IQ9.EQ.0) THEN
  WRITE(8,2221) J,I,STSUM
2221  FORMAT(1X,' AT J =',I2,' BAND ',I2,' QR =',D11.4)
ENDIF
145 SUM=SUM+STSUM
EOLD=E2I
150 CONTINUE
WRITE(8,103) I,SUM
103  FORMAT(/,' FOR BAND ',I2,' QR = ',D11.4,' WATTS/SQ.CM.')
WQR=WQR+SUM
180 CONTINUE
WRITE(8,101) WQR
101  FORMAT(/' TOTAL QR = ',D11.4,' WATTS/SQ.CM.')
200 CONTINUE
RETURN
END

```

```

C
C-----
C
SUBROUTINE FEI(E1I,E2I,E3I,Z)
IMPLICIT REAL*8(A-H,O-Z)
COMMON /RADC/ YBDY(100),B(100,8,40),OPTL(100,8,40)
CALL EXPI(Z,E1I,AUX)
E2I=DEXP(-Z)-Z*E1I
E3I=(DEXP(-Z)-Z*E2I)/2.0
C   E2I=DEXP(-Z)*DSQRT(3.D+00))
RETURN
END

```

```

C
C-----
C
SUBROUTINE EXPI(X,RES,AUX)
IMPLICIT REAL*8(A-H,O-Z)
COMMON /RADC/ YBDY(100),B(100,8,40),OPTL(100,8,40)
IF(X-1.) 2,1,1
1  YY=1./X
  AUX=1.-YY*(((YY+3.377358)*YY+2.052156)*YY+2.709479D-01)/((((YY*
C1.072553+5.716943)*YY+6.945239)*YY+2.593888)*YY+2.709496D-01)
  RES=AUX*YY*DEXP(-X)
  RETURN
2  IF(X+3.) 6,6,3
3  AUX=((((((7.122452D-7*X-1.766345D-6)*X+2.928433D-5)*X-2.335379D-4
C)*X+1.664156D-3)*X-1.041576D-2)*X+5.555682D-2)*X-2.500001D-1)*X
C+9.999999D-1
  RES=-1.D+30

```

```

      IF(X) 4,5,4
4     RES=X*AUX-DLOG(ABS(X))-5.772157D-1
5     RETURN
6     IF(X+9.) 8,8,7
7     AUX=1.-((((5.176245D-2*X+3.061037)*X+3.243655D+1)*X+2.244234D+2)*X
      C+2.486697D+2)/((((X+3.995161)*X+3.893944D+1)*X+2.263818D+1)*X
      C+1.807837D+2)
      GD TO 9
8     YY=9./X
      AUX=1.-YY*((((YY+7.659824D-1)*YY-7.271015D-1)*YY-1.080693)/((((YY
      C*2.518750+1.122927D+1)*YY+5.921405)*YY-8.666702)*YY-9.724216)
9     RES=AUX*DEXP(-X)/X
      RETURN
      END

```

C
C
C

```

      SUBROUTINE CARRAD (IQ4,IQ5,IQ9)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /A1/ TE1,TE2,M,IX,IMAX,IPSI,MODEL
      COMMON /A3/ EVIINF(25),THETAI(25),MUI(25),FI(25),DELHI(25),
      *      CIINF(25),LI(25)
      COMMON /A5/ EPSIIL(20,25),GIL(20,25),CID(200,50)
      COMMON /A9/ EVIS(25),XPST(100)
      COMMON /RAD/ RCON(100,10,40),RRHO(100,40),TEMP(100,40)
      *      ,TSTAG,Y(100,40),NXCON,NPTS(100)
      COMMON /RADC/ YBDY(100),B(100,8,40),OPTL(100,8,40)
      COMMON /RADO/ TEMPO(100,40),ALPN(100,40),ALPD(100,40)
      REAL*8 KAPPA(100,8,40),MUI
      WRITE(8,1)
1     FORMAT('///' CARLSON RADIATION MODEL')
      C      NUMBER DENSITIES
      IF((IQ4.EQ.5).OR.(IQ4.EQ.6)) GO TO 1111
      DO 12 K=3,NXCON
      DO 12 J=1,NPTS(K)
      DO 10 II=1,IMAX
10    RCON(K,II,J)=RRHO(K,J)*6.023D+23*RCON(K,II,J)/MUI(II)
12    CONTINUE

```

C

```

      C      NONEQUILIBRIUM RADIATION CORRECTION
      IF (IQ5.EQ.0) THEN
      DO 122 K=3,NXCON
      DO 122 J=1,NPTS(K)
      IF(J.EQ.NPTS(K)) THEN
      ALPN(K,J)=0.D+00
      GO TO 122
      ENDIF
      Q1=0.D+0
      Q2=0.D+0
      Q3=0.D+0
      DO 112 III=1,LI(3)
112   Q1=Q1+GIL(III,3)*DEXP(-EPSIIL(III,3)/TEMP(K,J))
      DO 113 III=1,LI(9)
113   Q2=Q2+GIL(III,9)*DEXP(-EPSIIL(III,9)/TEMP(K,J))
      DO 130 III=1,LI(1)
130   Q3=Q3+GIL(III,1)*DEXP(-EPSIIL(III,1)/TEMP(K,J))
      RHO1=RCON(K,3,J)*1.401D+01/6.023D+23
      RHO2=RCON(K,9,J)*1.401D+01/6.023D+23
      RHO3=RCON(K,1,J)*2.802D+01/6.023D+23
      RHO4=RHO1+RHO2+RHO3
      QVIB=1.D+0/(1.D+0-DEXP(-3.39D+03/TEMP(K,J)))
      QROT=TEMP(K,J)/5.8D+0
      C1=RHO4*2.4701D+01*QVIB*QROT*Q3*DEXP(1.135D+05/TEMP(K,J))
      C1=C1/(Q1**2*TEMP(K,J)**1.5)
      C2=RHO4*8.909D+06*Q1*DEXP(1.69D+05/TEMP(K,J))
      C2=C2/(Q2*TEMP(K,J)**1.5)
      ALPHA=0.5D+0
      BETA=0.5D+0
132   F=BETA**2*(1.D+0-ALPHA)**2*C1+BETA-1.D+0
      FA=-2.D+0*BETA**2*(1.D+0-ALPHA)*C1
      FB=2.D+0*BETA*(1.D+0-ALPHA)**2*C1+1.D+0
      G=ALPHA**2*BETA*C2+ALPHA-1.D+0
      GA=2.D+0*ALPHA*BETA*C2+1.D+0
      GB=ALPHA**2*C2
      DENOM=FA*GB-FB*GA
      DA=(-F*GB+G*FB)/DENOM
      DB=(-G*FA+F*GA)/DENOM
      ALPHA=ALPHA+DA
      BETA=BETA+DB
      IF((DABS(DA).GT.1.D-05).OR.(DABS(DB).GT.1.D-5)) GO TO 132
      ALPH=RCON(K,9,J)/(RCON(K,9,J)+RCON(K,3,J))

```

```

      BET=(RCON(K,9,J)+RCON(K,3,J))/(RCON(K,9,J)+RCON(K,3,J)
      *      +2.D+O*RCON(K,1,J))
      ALPN(K,J)=(ALPH**2/(1.D+O-ALPH))*((1.D+O-ALPH)/(ALPH**2))
      *      *(BET/BETA)
122  CONTINUE
      ENDIF
1111 CONTINUE
C
      PLANCK FUNCTION
      DO 13 K=3,NXCON
      DO 13 J=1,NPTS(K)
      TTT=TEMP(K,J)
      CALL PLANCK(B(K,1,J),B(K,2,J),B(K,3,J),B(K,4,J),B(K,5,J),TTT)
13  IF(IQ5.EQ.O) B(K,1,J)=ALPN(K,J)*B(K,1,J)
C
      ABSORPTION COEFFICIENTS
      DO 20 K=3,NXCON
      DO 20 J=1,NPTS(K)
      DO 20 I=1,5
      KAPPA(K,I,J)=ABSORB(K,J,I)
      IF((IQ5.EQ.O).AND.(I.NE.1)) KAPPA(K,I,J)=KAPPA(K,I,J)
      *      *ALPN(K,J)
20  CONTINUE
C
      TAU'S
      DO 100 K=3,NXCON
      DO 100 I=1,5
      OPTL(K,I,1)=KAPPA(K,I,1)*DABS(Y(K,1)-YBDY(K))
      DO 100 J=2,NPTS(K)
      OPTL(K,I,J)=OPTL(K,I,J-1)+(O.5*(KAPPA(K,I,J)+KAPPA(K,I,J-1))*
      *      DABS(Y(K,J-1)-Y(K,J)))
C
100 CONTINUE
C
      QRW
      DO 200 K=3,NXCON
      WRITE(8,102) XPST(K)
102  FORMAT(/,' AT X = ',D11.4)
      WQR=O.O
      DO 180 I=1,5
      CALL FEI(E1I,E2I,E3I,OPTL(K,I,1))
      EOLD=E2I
      JIN=1
      SUM=3.1415927*B(K,I,1)*EOLD*KAPPA(K,I,1)*DABS(Y(K,1)-YBDY(K))
      IF(IQ9.EQ.O) WRITE(8,2221) JIN,I,SUM
      DO 150 J=2,NPTS(K)
      Z=OPTL(K,I,J)
      IF(Z.EQ.O.D+O) THEN
      E2I=1.D+O
      GO TO 145
      ENDIF
      CALL FEI(E1I,E2I,E3I,Z)
      STSUM=3.14159*(B(K,I,J)*KAPPA(K,I,J)*E2I+B(K,I,J-1)*
      *      KAPPA(K,I,J-1)*EOLD)*DABS(Y(K,J-1)-Y(K,J))
      IF(IQ9.EQ.O) THEN
      WRITE(8,2221) J,I,STSUM
2221  FORMAT(1X,' AT J = ',I2,' BAND ',I2,' QR = ',D11.4)
      ENDIF
145  SUM=SUM+STSUM
      EOLD=E2I
150  CONTINUE
      WRITE(8,103) I,SUM
103  FORMAT(/' FOR BAND ',I2,' QR = ',D11.4,' WATTS/SQ.CM.')
      WQR=WQR+SUM
180  CONTINUE
      WRITE(8,101) WQR
101  FORMAT(/' TOTAL QR=',D11.4,' WATTS/SQ.CM.')
200  CONTINUE
      RETURN
      END

```

```

C
C-----
C
      SUBROUTINE PLANCK(B1,B2,B3,B4,B5,YY)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /RAD/ RCON(100,10,40),RRHO(100,40),TEMP(100,40)
      *      ,TSTAG,Y(100,40),NXCON,NPTS(100)
      COMMON /RADC/ YBDY(100),B(100,8,40),OPTL(100,8,40)
      T=YY
      IF(T.LE.20000.O)GOTO1027
1027 IF(T.GE.8000.O)GOTO1029
      B1=O.O
      B2=O.O
      B3=O.O
      B4=O.O

```



```

      B5=0.0
      RETURN
1029 IF(T.LE.10000.0)GOTO1030
      IF(T.LE.12000.0)GOTO1031
      IF(T.LE.14000.0)GOTO1032
      IF(T.LE.16000.0)GOTO1033
      IF(T.LE.18000.0)GOTO1034
      GOTO1240
1030 B1=10.0**(-65.26966+15.87146*DLOG10(T))
      B2=10.0**(-54.11981+13.22561*DLOG10(T))
      B3=10.0**(-44.83075+11.05391*DLOG10(T))
      B4=10.0**(-16.52511+4.425444*DLOG10(T))
      B5=10.0**(-8.874156+2.216357*DLOG10(T))
      B1=B1*1000.00
      B2=B2*1000.00
      B3=B3*1000.00
      B4=B4*1000.00
      B5=B5*1000.00
      RETURN
1031 B1=(10.0**(-54.67501+13.22279*DLOG10(T)))*1000.0
      B2=(10.0**(-44.60436+10.84674*DLOG10(T)))*1000.0
      B3=(10.0**(-36.91518+9.075018*DLOG10(T)))*1000.0
      B4=(10.0**(-15.30031+4.119244*DLOG10(T)))*1000.0
      B5=(10.0**(-7.831680+1.955738*DLOG10(T)))*1000.0
      RETURN
1032 B1=(10.0**(-47.32488+11.42093*DLOG10(T)))*1000.0
      B2=(10.0**(-37.87564+9.197217*DLOG10(T)))*1000.0
      B3=(10.0**(-31.30212+7.698991*DLOG10(T)))*1000.0
      B4=(10.0**(-14.17392+3.843113*DLOG10(T)))*1000.0
      B5=(10.0**(-7.128949+1.783465*DLOG10(T)))*1000.0
      RETURN
1033 B1=(10.0**(-41.90448+10.11359*DLOG10(T)))*1000.0
      B2=(10.0**(-32.87168+7.990316*DLOG10(T)))*1000.0
      B3=(10.0**(-27.12177+6.690739*DLOG10(T)))*1000.0
      B4=(10.0**(-13.12000+3.588921*DLOG10(T)))*1000.0
      B5=(10.0**(-6.625956+1.662149*DLOG10(T)))*1000.0
      RETURN
1034 B1=(10.0**(-37.77432+9.131181*DLOG10(T)))*1000.0
      B2=(10.0**(-28.96032+7.059955*DLOG10(T)))*1000.0
      B3=(10.0**(-23.90936+5.926629*DLOG10(T)))*1000.0
      B4=(10.0**(-12.13764+3.355253*DLOG10(T)))*1000.0
      B5=(10.0**(-6.259400+1.574959*DLOG10(T)))*1000.0
      RETURN
1240 B1=(10.0**(-34.49941+8.36166*DLOG10(T)))*1000.0
      B2=(10.0**(-25.87330+6.33544*DLOG10(T)))*1000.0
      B3=(10.0**(-21.34066+5.32298*DLOG10(T)))*1000.0
      B4=(10.0**(-11.28150+3.1540*DLOG10(T)))*1000.0
      B5=(10.0**(-5.97387+1.50786*DLOG10(T)))*1000.0
      RETURN
      END

```

C
C-----
C

```

      FUNCTION ABSORB(K,I,ISTEP)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /RAD/ RCON(100,10,40),RRHO(100,40),TEMP(100,40)
      *      ,TSTAG,Y(100,40),NXCON,NPTS(100)
      COMMON /RADC/ YBDY(100),B(100,8,40),OPTL(100,8,40)
C      THIS PROGRAM USES 5 WAVELENGTH STEPS
C      THE NEXT CARD MUST BE CHANGED FOR GRAY GAS CASE
      IGRAY=0.0
      T=TEMP(K,I)
      BLACKB=((5.6697D-05)*(T**4))/3.14159
      YYY=T
      YY=T/10000.0
      IF(YYY.LE.20000.0)GOTO1035
1036 FORMAT(1H0,5X,'T TEST FAILED IN ABSORB')
      CONTINUE
1035 IF(IGRAY.EQ.1)GO TO 1037
      GO TO (1037,1038,1039,1040,1041),ISTEP
1037 IF(YYY.LT.8000.0)GO TO 1042
      IF(YYY.LE.10000.0)GOTO1043
      IF(YYY.LE.12000.0)GOTO1044
      IF(YYY.LE.14000.0)GOTO1045
      IF(YYY.LE.16000.0)GOTO1046
      IF(YYY.LE.18000.0)GOTO1047
      GOTO1241
1042 ABSORB=0.0
      GO TO 1048
1043 ABSORB=(1.0061D-13)*(10.0**(-2.6064*YY))

```

```

GO TO 1049
1044 ABSORB=(9.6882D-15)*(10.0**(-1.5953*YY))
GO TO 1049
1045 ABSORB=(2.0340D-15)*(10.0**(-1.0304*YY))
GO TO 1049
1046 ABSORB=(6.9685D-16)*(10.0**(-0.6981*YY))
GO TO 1049
1047 ABSORB=(3.3188D-16)*(10.0**(-0.49675*YY))
GOTO 1049
1241 ABSORB=(2.0976D-16)*(10.0**(-0.38605*YY))
1049 FNDEN=RCON(K,3,I)
EDEKT=(4.707D-06)*(RCON(K,7,I)**(2.0/7.0))/(((0.86176D-04)*YYY)**
11.5)
EDEKT=DEXP(EDEKT)
ABSORB=(ABSORB*EDEKT)*FNDEN
1048 ABSOR1=ABSORB
IF (IGRAY.EQ.1)GOTO 1038
RETURN
1038 IF(YYY.LT.8000.0)GOTO 1050
IF(YYY.LE.10000.0)GOTO 1051
IF(YYY.LE.12000.0)GOTO 1052
IF(YYY.LE.14000.0)GOTO 1053
IF(YYY.LE.16000.0)GOTO 1054
IF(YYY.LE.18000.0)GOTO 1055
GOTO 1242
1050 ABSORB=0.0
GOTO 1056
1051 ABSORB=(5.9872D-22)*(10.0**(2.01935*YY))
GOTO 1057
1052 ABSORB=(3.2660D-21)*(10.0**(1.28255*YY))
GOTO 1057
1053 ABSORB=(1.1243D-20)*(10.0**(0.83515*YY))
GOTO 1057
1054 ABSORB=(2.1562D-20)*(10.0**(0.63315*YY))
GOTO 1057
1055 ABSORB=(3.7012D-20)*(10.0**(0.4865*YY))
GOTO 1057
1242 ABSORB=(5.4757D-20)*(10.0**(0.3924*YY))
1057 FNDEN=RCON(K,3,I)
EDEKT=(4.707D-06)*(RCON(K,7,I)**(2.0/7.0))/(((0.86176D-04)*YYY)**
11.5)
EDEKT=DEXP(EDEKT)
ABSORB=(ABSORB*EDEKT)*FNDEN
1056 ABSOR2=ABSORB
IF (IGRAY.EQ.1)GOTO 1039
RETURN
1039 IF(YYY.LT.8000.0)GOTO 1058
IF(YYY.LE.10000.0)GOTO 1059
IF(YYY.LE.12000.0)GOTO 1060
IF(YYY.LE.14000.0)GOTO 1061
IF(YYY.LE.16000.0)GOTO 1062
IF(YYY.LE.18000.0)GOTO 1063
GOTO 1243
1058 ABSORB=0.0
GOTO 1064
1059 ABSORB=((7.266D-20)*(10.0**(21.267*YY)))*(1.0D-20)
GOTO 1065
1060 ABSORB=(2.2772D-19)*(10.0**(0.77045*YY))
GOTO 1065
1061 ABSORB=(4.8471D-19)*(10.0**(0.49705*YY))
GOTO 1065
1062 ABSORB=(8.1764D-19)*(10.0**(0.33485*YY))
GOTO 1065
1063 ABSORB=(1.1902D-18)*(10.0**(0.23295*YY))
GOTO 1065
1243 ABSORB=(2.19875D-18)*(10.0**(0.08485*YY))
1065 FNDEN=RCON(K,3,I)
ABSORB=ABSORB*FNDEN
1064 ABSOR3=ABSORB
IF (IGRAY.EQ.1)GOTO 1040
RETURN
1040 IF(YYY.LT.8000.0)GOTO 1066
IF(YYY.LE.10000.0)GOTO 1067
IF(YYY.LE.12000.0)GOTO 1068
IF(YYY.LE.14000.0)GOTO 1069
IF(YYY.LE.16000.0)GOTO 1070
IF(YYY.LE.18000.0)GOTO 1071
GOTO 1244
1066 ABSORB=0.0
GOTO 1072

```

```

1067 ABSORB=(3.4065D-30)*(10.0**(7.2773*YY))
      GOTO1073
1068 ABSORB=(8.3188D-28)*(10.0**(4.8895*YY))
      GOTO1073
1069 ABSORB=(2.8643D-26)*(10.0**(3.6087*YY))
      GOTO1073
1070 ABSORB=(1.9928D-25)*(10.0**(3.00695*YY))
      GOTO1073
1071 ABSORB=(3.9376D-24)*(10.0**(2.1971*YY))
      GOTO1073
1244 ABSORB=(4.2515D-24)*(10.0**(2.1786*YY))
1073 FNDEN=RCON(K,3,I)
      EDEKT=(4.707D-06)*(RCON(K,7,I)**(2.0/7.0))/(((0.86176D-04)*YYY)**
11.5)
      EDEKT=DEXP(EDEKT)
      ABSORB=(ABSORB*EDEKT)*FNDEN
1072 ABSOR4=ABSORB
      IF(IGRAY.EQ.1)GOTO1041
      RETURN
1041 IF(YYY.LT.8000.0)GOTO1074
      IF(YYY.LE.10000.0)GOTO1075
      IF(YYY.LE.12000.0)GOTO1076
      IF(YYY.LE.14000.0)GOTO1077
      IF(YYY.LE.16000.0)GOTO1078
      IF(YYY.LE.18000.0)GOTO1079
      GOTO1245
1074 ABSORB=0.0
      GOTO1080
1075 ABSORB=(3.1595D-27)*(10.0**(6.54335*YY))
      GOTO1081
1076 ABSORB=(6.8837D-25)*(10.0**(4.20515*YY))
      GOTO1081
1077 ABSORB=(1.8079D-23)*(10.0**(3.02235*YY))
      GOTO1081
1078 ABSORB=(4.1675D-22)*(10.0**(2.04900*YY))
      GOTO1081
1079 ABSORB=(2.4630D-21)*(10.0**(1.56667*YY))
      GOTO1081
1245 ABSORB=(1.0535D-20)*(10.0**(1.2161*YY))
1081 FNDEN=RCON(K,3,I)
      ABSORB=ABSORB*FNDEN
1080 ABSOR5=ABSORB
      IF(IGRAY.EQ.1)GOTO1082
      RETURN
1082 ABSORB=(ABSOR1*B(K,1,I)+ABSOR2*B(K,2,I)+ABSOR3*B(K,3,I)+
1ABSOR4*B(K,4,I)+ABSOR5*B(K,5,I))/BLACKB
      ABSORB=ABSORB*(1.0D+07)
      RETURN
      END

```

C
C
C

```

SUBROUTINE CORRAD (IQ4,IQ5,IQ6,IQ8,IQ9)
IMPLICIT REAL*8(A-H,O-Z)
COMMON /A1/ TE1,TE2,M,IX,IMAX,IPSI,MODEL
COMMON /A3/ EVIINF(25),THETAI(25),MUI(25),FI(25),DELHI(25),
* CIINF(25),LI(25)
COMMON /A5/ EPSIIL(20,25),GIL(20,25),CID(200,50)
COMMON /A9/ EVIS(25),XPST(100)
COMMON /RAD/ RCON(100,10,40),RRHO(100,40),TEMP(100,40)
* ,TSTAG,Y(100,40),NXCON,NPTS(100)
COMMON /RADC/ YBDY(100),B(100,8,40),OPLT(100,8,40)
COMMON /RADO/ TEMPO(100,40),ALPN(100,40),ALPD(100,40)
COMMON /RADDL/ BETN(100,40),BETO(100,40)
DIMENSION CR(40,13)
REAL*8 KAPPA(100,8,40),MUI
WRITE(8,1)
1 FORMAT(///' CORNELL RADIATION MODEL')
IF(IQ8.EQ.0) THEN
  IF(IQ6.EQ.0) WRITE(8,2)
  IF(IQ6.EQ.1) WRITE(8,3)
ENDIF
2 FORMAT('/' BETA=BETA(Te)')
3 FORMAT('/' BETA=BETA(Tt)')

```

C
C

NUMBER DENSITIES
AND NONEQUILIBRIUM CORRECTION

```

IF((IQ4.EQ.5).OR.(IQ4.EQ.6)) GO TO 1111
DO 10 K=3,NXCON
DO 10 J=1,NPTS(K)
DO 10 II=1,IMAX

```

```

XXX=RRHO(K,J)*6.023D+23
XXX=XXX*RCON(K,II,J)/MUI(II)
10 RCON(K,II,J)=XXX
IF (IQ5.EQ.O) THEN
  DO 122 K=3,NXCON
  DO 122 J=1,NPTS(K)
  IF(J.EQ.NPTS(K)) THEN
    ALPN(K,J)=O.D+OO
    BETN(K,J)=O.D+OO
    GO TO 122
  ENDIF
  Q1=O.D+O
  Q2=O.D+O
  Q3=O.D+O
  DO 112 III=1,LI(3)
112 Q1=Q1+GIL(III,3)*DEXP(-EPSIIL(III,3)/TEMP(K,J))
  DO 113 III=1,LI(9)
113 Q2=Q2+GIL(III,9)*DEXP(-EPSIIL(III,9)/TEMP(K,J))
  DO 130 III=1,LI(1)
130 Q3=Q3+GIL(III,1)*DEXP(-EPSIIL(III,1)/TEMP(K,J))
  RHO1=RCON(K,3,J)*1.401D+O1/6.023D+23
  RHO2=RCON(K,9,J)*1.401D+O1/6.023D+23
  RHO3=RCON(K,1,J)*2.802D+O1/6.023D+23
  RHO4=RHO1+RHO2+RHO3
  QVIB=1.D+O/(1.D+O-DEXP(-3.39D+O3/TEMP(K,J)))
  QROT=TEMP(K,J)/5.8D+O
  C1=RHO4*2.4701D+O1*QVIB*QROT*Q3*DEXP(1.135D+O5/TEMP(K,J))
  C1=C1/(Q1**2*TEMP(K,J)**1.5)
  C2=RHO4*8.909D+O6*Q1*DEXP(1.69D+O5/TEMP(K,J))
  C2=C2/(Q2*TEMP(K,J)**1.5)
  ALPHA=O.5D+O
  BETA=O.5D+O
132 F=BETA**2*(1.D+O-ALPHA)**2*C1+BETA-1.D+O
  FA=-2.D+O*BETA**2*(1.D+O-ALPHA)*C1
  FB=2.D+O*BETA*(1.D+O-ALPHA)**2*C1+1.D+O
  G=ALPHA**2*BETA*C2+ALPHA-1.D+O
  GA=2.D+O*ALPHA*BETA*C2+1.D+O
  GB=ALPHA**2*C2
  DENOM=FA*GB-FB*GA
  DA=(-F*GB+G*FB)/DENOM
  DB=(-G*FA+F*GA)/DENOM
  ALPHA=ALPHA+DA
  BETA=BETA+DB
  IF((DABS(DA).GT.1.D+O5).OR.(DABS(DB).GT.1.D+O5)) GO TO 132
  ALPH=RCON(K,9,J)/(RCON(K,9,J)+RCON(K,3,J))
  BET=(RCON(K,9,J)+RCON(K,3,J))/(2.D+O*RCON(K,1,J)+
  * RCON(K,9,J)+RCON(K,3,J))
  IF(IQ6.EQ.1) GO TO 1888
  BETN(K,J)=(BET**2/(1.D+O-BET))*((1.D+O-BETA)/(BETA**2))
  * ((1.D+O-ALPH)**2/(1.D+O-ALPHA)**2)
1888 ALPN(K,J)=BET*(ALPH**2/(1.D+O-ALPH))
  * ((1.D+O-ALPHA)/(BETA*ALPHA**2))
122 CONTINUE
  DO 142 K=3,NXCON
  DO 142 J=1,NPTS(K)
  IF(J.EQ.NPTS(K)) THEN
    ALPO(K,J)=O.D+OO
    BETO(K,J)=O.D+OO
    GO TO 142
  ENDIF
  Q1=O.D+O
  Q2=O.D+O
  Q3=O.D+O
  DO 152 III=1,LI(4)
152 Q1=Q1+GIL(III,4)*EXP(-EPSIIL(III,4)/TEMP(K,J))
  DO 153 III=1,LI(10)
153 Q2=Q2+GIL(III,10)*DEXP(-EPSIIL(III,10)/TEMP(K,J))
  DO 160 III=1,LI(2)
160 Q3=Q3+GIL(III,2)*DEXP(-EPSIIL(III,2)/TEMP(K,J))
  RHO1=RCON(K,4,J)*1.600D+O1/6.023D+23
  RHO2=RCON(K,10,J)*1.600D+O1/6.023D+23
  RHO3=RCON(K,2,J)*3.200D+O1/6.023D+23
  RHO4=RHO1+RHO2+RHO3
  QVIB=1.D+O/(1.D+O-DEXP(-2.27D+O3/TEMP(K,J)))
  QROT=TEMP(K,J)/4.2D+O
  C1=RHO4*1.7725D+O1*QVIB*QROT*Q3*DEXP(5.950D+O4/TEMP(K,J))
  C1=C1/(Q1**2*TEMP(K,J)**1.5)
  C2=RHO4*7.801D+O6*Q1*DEXP(1.58D+O5/TEMP(K,J))
  C2=C2/(Q2*TEMP(K,J)**1.5)
  ALPHA=O.5D+O

```

```

BETA=0.5D+0
162 F=BETA**2*(1.D+O-ALPHA)**2*C1+BETA-1.D+O
FA=-2.D+O*BETA**2*(1.D+O-ALPHA)*C1
FB=2.D+O*BETA*(1.D+O-ALPHA)**2*C1+1.D+O
G=ALPHA**2*BETA*C2+ALPHA-1.D+O
GA=2.D+O*ALPHA*BETA*C2+1.D+O
GB=ALPHA**2*C2
DENOM=FA*GB-FB*GA
DA=(-F*GB+G*FB)/DENOM
DB=(-G*FA+F*GA)/DENOM
ALPHA=ALPHA+DA
BETA=BETA+DB
IF((DABS(DA).GT.1.D-05).OR.(DABS(DB).GT.1.D-5)) GO TO 162
ALPH=RCON(K,10,J)/(RCON(K,10,J)+RCON(K,4,J))
BET=(RCON(K,10,J)+RCON(K,4,J))/(2.D+O*RCON(K,2,J)+
* RCON(K,10,J)+RCON(K,4,J))
IF(IQ6.EQ.1) GO TO 1999
BETO(K,J)=(BET**2/(1.D+O-BET))*((1.D+O-BETA)/(BETA**2))
* ((1.D+O-ALPH)**2/(1.D+O-ALPHA)**2)
1999 ALPO(K,J)=BET*(ALPH**2/(1.D+O-ALPH))
* ((1.D+O-ALPHA)/(BETA*ALPHA**2))
142 CONTINUE
C BETA=BETA(Tt)
IF(IQ6.EQ.1) THEN
DO 1122 K=3,NXCON
DO 1122 J=1,NPTS(K)
IF(J.EQ.NPTS(K)) THEN
BETN(K,J)=O.D+OO
GO TO 1122
ENDIF
Q1=O.D+O
Q2=O.D+O
Q3=O.D+O
DO 1112 III=1,LI(3)
1112 Q1=Q1+GIL(III,3)*DEXP(-EPSIIL(III,3)/TEMPO(K,J))
DO 1113 III=1,LI(9)
1113 Q2=Q2+GIL(III,9)*DEXP(-EPSIIL(III,9)/TEMPO(K,J))
DO 1130 III=1,LI(1)
1130 Q3=Q3+GIL(III,1)*DEXP(-EPSIIL(III,1)/TEMPO(K,J))
RH01=RCON(K,3,J)*1.401D+01/6.023D+23
RH02=RCON(K,9,J)*1.401D+01/6.023D+23
RH03=RCON(K,1,J)*2.802D+01/6.023D+23
RH04=RH01+RH02+RH03
QVIB=1.D+O/(1.D+O-DEXP(-3.39D+03/TEMPO(K,J)))
QROT=TEMPO(K,J)/5.8D+O
C1=RH04*2.4701D+01*QVIB*QROT*Q3*DEXP(1.135D+05/TEMPO(K,J))
C1=C1/(Q1**2*TEMPO(K,J)**1.5)
C2=RH04*8.909D+06*Q1*DEXP(1.69D+05/TEMPO(K,J))
C2=C2/(Q2*TEMPO(K,J)**1.5)
ALPHA=O.5D+O
BETA=O.5D+O
1132 F=BETA**2*(1.D+O-ALPHA)**2*C1+BETA-1.D+O
FA=-2.D+O*BETA**2*(1.D+O-ALPHA)*C1
FB=2.D+O*BETA*(1.D+O-ALPHA)**2*C1+1.D+O
G=ALPHA**2*BETA*C2+ALPHA-1.D+O
GA=2.D+O*ALPHA*BETA*C2+1.D+O
GB=ALPHA**2*C2
DENOM=FA*GB-FB*GA
DA=(-F*GB+G*FB)/DENOM
DB=(-G*FA+F*GA)/DENOM
ALPHA=ALPHA+DA
BETA=BETA+DB
IF((DABS(DA).GT.1.D-05).OR.(DABS(DB).GT.1.D-5)) GO TO 1132
BET=(RCON(K,9,J)+RCON(K,3,J))/(2.D+O*RCON(K,1,J)+
* RCON(K,9,J)+RCON(K,3,J))
ALPH=RCON(K,9,J)/(RCON(K,9,J)+RCON(K,3,J))
BETN(K,J)=(BET**2/(1.D+O-BET))*((1.D+O-BETA)/(BETA**2))
* ((1.D+O-ALPH)**2/(1.D+O-ALPHA)**2)
1122 CONTINUE
DO 1142 K=3,NXCON
DO 1142 J=1,NPTS(K)
IF(J.EQ.NPTS(K)) THEN
BETO(K,J)=O.D+OO
GO TO 1142
ENDIF
Q1=O.D+O
Q2=O.D+O
Q3=O.D+O
DO 1152 III=1,LI(4)
1152 Q1=Q1+GIL(III,4)*EXP(-EPSIIL(III,4)/TEMPO(K,J))

```

```

DO 1153 III=1,LI(10)
1153 Q2=Q2+GIL(III,10)*DEXP(-EPSIIL(III,10)/TEMPO(K,J))
DO 1160 III=1,LI(2)
1160 Q3=Q3+GIL(III,2)*DEXP(-EPSIIL(III,2)/TEMPO(K,J))
RH01=RCON(K,4,J)*1.600D+01/6.023D+23
RH02=RCON(K,10,J)*1.600D+01/6.023D+23
RH03=RCON(K,2,J)*3.200D+01/6.023D+23
RH04=RH01+RH02+RH03
QVIB=1.D+O/(1.D+O-DEXP(-2.27D+03/TEMPO(K,J)))
QROT=TEMPO(K,J)/4.2D+O
C1=RH04*1.7725D+01*QVIB*QROT*Q3*DEXP(5.950D+04/TEMPO(K,J))
C1=C1/(Q1**2*TEMPO(K,J)**1.5)
C2=RH04*7.801D+06*Q1*DEXP(1.58D+05/TEMPO(K,J))
C2=C2/(Q2*TEMPO(K,J)**1.5)
ALPHA=O.5D+O
BETA=O.5D+O
1162 F=BETA**2*(1.D+O-ALPHA)**2*C1+BETA-1.D+O
FA=-2.D+O*BETA**2*(1.D+O-ALPHA)*C1
FB=2.D+O*BETA*(1.D+O-ALPHA)**2*C1+1.D+O
G=ALPHA**2*BETA*C2+ALPHA-1.D+O
GA=2.D+O*ALPHA*BETA*C2+1.D+O
GB=ALPHA**2*C2
DENOM=FA*GB-FB*GA
DA=(-F*GB+G*FB)/DENOM
DB=(-G*FA+F*GA)/DENOM
ALPHA=ALPHA+DA
BETA=BETA+DB
IF((DABS(DA).GT.1.D+O5).OR.(DABS(DB).GT.1.D+O5)) GO TO 1162
BET=(RCON(K,10,J)+RCON(K,4,J))/(2.D+O*RCON(K,2,J)+
* RCON(K,10,J)+RCON(K,4,J))
ALPH=RCON(K,10,J)/(RCON(K,10,J)+RCON(K,4,J))
BETO(K,J)=(BET**2/(1.D+O-BET))*((1.D+O-BETA)/(BETA**2))
* ((1.D+O-ALPH)**2/(1.D+O-ALPHA)**2)
1142 CONTINUE
ENDIF
ENDIF
1111 CONTINUE
C QRW
DO 200 K=3,NXCON
WRITE(8,102) XPST(K)
102 FORMAT(///,' AT X = ',D11.4)
WQR=O.O
DO 120 J=1,NPTS(K)
TT=TEMP(K,J)
CR(J,1)=3.6D-13*RCON(K,2,1)*DEXP(-5.7D+4/TT)
CR(J,2)=1.8D-14*RCON(K,1,1)*DEXP(-8.3D+4/TT)
CR(J,3)=3.0D-13*RCON(K,5,1)*DEXP(-6.7D+4/TT)
CR(J,4)=2.1D-19*RCON(K,5,1)*DEXP(-7.6D+3/TT)
SM=2.0D-20*(RCON(K,3,1)+RCON(K,7,1))*DEXP(-5.2D+04/TT)
CR(J,5)=SM*1.0D-14
SM=3.0D-20*(RCON(K,4,1)+RCON(K,7,1))*DEXP(-4.2D+04/TT)
CR(J,6)=SM*1.0D-14
IF(IQ5.EQ.O) THEN
CR(J,7)=1.D-10*RCON(K,3,J)*DEXP(-1.8D+05/TT)*ALPN(K,J)
CR(J,8)=8.D-13*RCON(K,3,J)*DEXP(-5.9D+04/TT)*ALPN(K,J)
CR(J,9)=9.D-11*RCON(K,4,J)*DEXP(-1.7D+05/TT)*ALPO(K,J)
CR(J,10)=8.D-13*RCON(K,4,J)*DEXP(-5.9D+04/TT)*ALPO(K,J)
IF(IQ8.EQ.O) THEN
CR(J,13)=1.6D-12*RCON(K,8,J)*DEXP(-3.6D+04/TT)*BETN(K,J)
ELSE
CR(J,13)=1.6D-12*RCON(K,8,J)*DEXP(-3.6D+04/TT)
ENDIF
ELSE
CR(J,7)=1.D-10*RCON(K,3,J)*DEXP(-1.8D+05/TT)
CR(J,8)=8.D-13*RCON(K,3,J)*DEXP(-5.9D+04/TT)
CR(J,9)=9.D-11*RCON(K,4,J)*DEXP(-1.7D+05/TT)
CR(J,10)=8.D-13*RCON(K,4,J)*DEXP(-5.9D+04/TT)
CR(J,13)=1.6D-12*RCON(K,8,J)*DEXP(-3.6D+04/TT)
ENDIF
CR(J,11)=9.D-33*(RCON(K,9,J)+RCON(K,7,J))
**DEXP(-1.4D+04/TT)
CR(J,12)=9.D-33*(RCON(K,10,J)+RCON(K,7,J))
**DEXP(-1.4D+04/TT)
120 CONTINUE
DO 180 I=1,13
JIN=1
RSUM=3.1415927D+00*CR(1,I)*DABS(Y(K,1)-YBDY(K))
IF(IQ9.EQ.O) WRITE(8,2221) JIN,I,RSUM
DO 150 J=2,NPTS(K)
STSUM=O.5D+00*(CR(J,I)+CR(J-1,I))*DABS(Y(K,J-1)-Y(K,J))

```



```

      BETA=BETA+DB
      IF((DABS(DA).GT.1.D-05).OR.(DABS(DB).GT.1.D-5)) GO TO 132
      ALPH=RCON(K,9,J)/(RCON(K,9,J)+RCON(K,3,J))
      BET=(RCON(K,9,J)+RCON(K,3,J))/(RCON(K,9,J)+RCON(K,3,J)
      *      +2.D+O*RCON(K,1,J))
      ALPN(K,J)=(ALPH**2/(1.D+O-ALPH))*((1.D+O-ALPHA)/(ALPHA**2))
      *      *(BET/BETA)
122  CONTINUE
      ENDIF
1111 CONTINUE
C
      DENSITY RATIOS
      DO 12 K=3,NXCON
      DO 12 J=1,NPTS(K)
      IF(IQ4.EQ.4) RCON(K,7,J)=RCON(K,7,J)*RRHO(K,J)*6.023D+23/MUI(7)
      RRHO(K,J)=RRHO(K,J)/1.225D-03
12  CONTINUE
C
      PLANCK FUNCTION
      DO 30 K=3,NXCON
      DO 30 J=1,NPTS(K)
      TT=TEMP(K,J)/168800.0
C
      IF(TEMP(K,J).LE.8.D+03) GO TO 30
      BLACK=(TT*168800.)*4*5.6696D-12/3.1415927
      FF=15.D+00/(3.1415927**4)
      X1=1./TT
      B1=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C      DEXP(-2.*X1)*0.5*(.75+1.5*X1+1.5*X1**2+X1**3))
      X1=0.935/TT
      B2=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C      DEXP(-2.*X1)*0.5*(.75+1.5*X1+1.5*X1**2+X1**3))
      X1=0.835/TT
      B3=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C      DEXP(-2.*X1)*0.5*(.75+1.5*X1+1.5*X1**2+X1**3))
      X1=0.754/TT
      B4=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C      DEXP(-2.*X1)*0.5*(.75+1.5*X1+1.5*X1**2+X1**3))
      X1=0.473/TT
      B6=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C      DEXP(-2.*X1)*0.5*(.75+1.5*X1+1.5*X1**2+X1**3))
      X1=0.213/TT
      B8=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C      DEXP(-2.*X1)*0.5*(.75+1.5*X1+1.5*X1**2+X1**3))
      B8=1-B8
      B2=B2-B1
      B3=B3-B2
      B4=B4-B3
      B6=B6-B4
      B5=1-B8-B6-B4-B3-B2-B1
      B(K,1,J)=BLACK*B1
      B(K,2,J)=BLACK*B2
      B(K,3,J)=BLACK*B3
      B(K,4,J)=BLACK*B4
      IF(IQ5.EQ.0) THEN
      B(K,1,J)=B(K,1,J)*ALPN(K,J)
      B(K,2,J)=B(K,2,J)*ALPN(K,J)
      B(K,3,J)=B(K,3,J)*ALPN(K,J)
      B(K,4,J)=B(K,4,J)*ALPN(K,J)
      ENDIF
      B(K,6,J)=BLACK*B6
      B(K,7,J)=BLACK*2.4D-21*RCON(K,7,J)*B6*DEXP(.162/TT)
      B(K,8,J)=BLACK*B8
      B(K,5,J)=BLACK*B5
      B(K,1,J)=B(K,1,J)+B(K,2,J)+B(K,3,J)+B(K,4,J)
      B(K,2,J)=B(K,5,J)+B(K,6,J)+B(K,7,J)+B(K,8,J)
30  CONTINUE
C
      ABSORPTION COEFFICIENTS
      DO 20 K=3,NXCON
      DO 20 J=1,NPTS(K)
      TT=TEMP(K,J)/1.0D+04
      IF(TEMP(K,J).LE.1.1D+04) THEN
      KAPPA(K,1,J)=36.D+00*RRHO(K,J)*TT**4.02
      ELSE
      KAPPA(K,1,J)=RRHO(K,J)*(8.1D+00+41.3D+00*TT)
      ENDIF
      IF(TEMP(K,J).LE.7.D+03) THEN
      AA=234.7D+00
      BB=1.41D+00
      CC=7.83D+00
      GO TO 25
      ENDIF
      IF(TEMP(K,J).LE.9.D+03) THEN

```



```

AA=262.D+00
BB=1.4D+00
CC=8.19D+00
GO TO 25
ENDIF
IF(TEMP(K,J).LE.1.1D+04) THEN
AA=266.D+00
BB=1.15D+00
CC=13.86D+00
GO TO 25
ENDIF
IF(TEMP(K,J).LE.1.3D+04) THEN
AA=232.D+00
BB=1.06D+00
CC=13.1D+00
GO TO 25
ENDIF
AA=406.5D+00
BB=1.1D+00
CC=11.3D+00
25 JP=AA*RRHO(K,J)**BB*TT**CC
IF(B(K,2,J).LE.1.D-05) THEN
KAPPA(K,2,J)=0.D+00
GO TO 20
ENDIF
KAPPA(K,2,J)=JP/B(K,2,J)
IF(IQ5.EQ.0) THEN
KAPPA(K,2,J)=KAPPA(K,2,J)*ALPN(K,J)
ENDIF
20 CONTINUE
C TAU'S
DO 100 K=3,NXCON
DO 100 I=1,2
OPTL(K,I,1)=5.D-01*KAPPA(K,I,1)*DABS(Y(K,1)-YBDY(K))
DO 100 J=2,NPTS(K)
OPTL(K,I,J)=OPTL(K,I,J-1)+(0.5*(KAPPA(K,I,J)+KAPPA(K,I,J-1))*
DABS(Y(K,J-1)-Y(K,J)))
C
100 CONTINUE
C QRW
DO 200 K=3,NXCON
WRITE(8,102) XPST(K)
102 FORMAT(///,' AT X = ',D11.4)
WQR=0.0
DO 180 I=1,2
CALL FEI(E1I,E2I,E3I,OPTL(K,I,1))
EOLD=E2I
JIN=1
SUM=3.1415927*EOLD*B(K,I,1)*KAPPA(K,I,1)*DABS(Y(K,1)-YBDY(K))
IF(IQ9.EQ.0) WRITE(8,2221) JIN,I,SUM
DO 150 J=2,NPTS(K)
Z=OPTL(K,I,J)
IF(Z.EQ.0.D+0) THEN
E2I=1.D+0
GO TO 145
ENDIF
CALL FEI(E1I,E2I,E3I,Z)
STSUM=3.14159*(B(K,I,J)*KAPPA(K,I,J)*E2I+B(K,I,J-1)*
KAPPA(K,I,J-1)*EOLD)*DABS(Y(K,J-1)-Y(K,J))
IF(IQ9.EQ.0) THEN
WRITE(8,2221) J,I,STSUM
2221 FORMAT(1X,' AT J = ',I2,' BAND ',I2,' QR = ',D11.4)
ENDIF
145 SUM=SUM+STSUM
EOLD=E2I
150 CONTINUE
WRITE(8,103) I,SUM
103 FORMAT(/,' FOR BAND ',I2,' QR = ',D11.4,' WATTS/SQ.CM.')
WQR=WQR+SUM
180 CONTINUE
WRITE(8,101) WQR
101 FORMAT(/' TOTAL QR = ',D11.4,' WATTS/SQ.CM.')
200 CONTINUE
IF((IQ4.EQ.5).OR.(IQ4.EQ.6)) THEN
DO 1000 K=3,NXCON
DO 1000 J=1,NPTS(K)
RRHO(K,J)=RRHO(K,J)*1.225D-03
1000 CONTINUE
ENDIF
RETURN
END

```

LISTING OF INPUT FILE SCREEN

```

*  --INPUT NAMELIST--
*
*  LINE #1
*  IPF  - OUTPUT PRINT FREQUENCY
*  M    - VIBRATIONAL EQUILIBRIUM (Y=0,N=1)
*  NSR  - NO. OF STRLINES IN THE STAGNATION REGION
*  IANS - FULL FLOW FIELD SOLN. (Y=0,N=1)
*  MW   - MILLIKAN & WHITE DATA (Y=1,N=2)
*  MODEL - VIBEQ=0, CVD=1, CVDV=2, CVDV-PREF=3, PARK=4
*  IQ3  - FROZEN FLOW=1, N2-FROZEN,O2-DISS=2, N2&O2-DISS=3
*  IQ1  - (TVIB=TINF)=1, (TVIB=TTRANS)=2 (FOR DISS. SHOCK JUMP)
*  IQ2  - (TEL=TINF)=1, (TEL=TTRANS)=2 (FOR DISS. SHOCK JUMP)
*  IQ4  - NO RAD.MODELS=0, OLSTAD=1, CARLSON=2, CORNELL=3, ANDERSON=4
*        ALL MODELS=5, ALL MODELS WITH CAPTURED VALUES=6
*  IQ5  - NON-EQUIL CORR. FOR ATOMIC RADIATION (Y=0,N=1)
*  IQ7  - STRLNE #. FOR DETAILS (NO STRLNE DETAIL=0)
*  IQ6  - (BETA=BETA(TE))=0,(BETA=BETA(T))=1
*  IQ8  - NON-EQUIL CORR. FOR MOLECULAR RADIATION (Y=0,N=1)
*  IQ9  - VALUES AT EACH STEP (Y=0,N=1)
*
*  LINE #2
*  IS1,IS2,IS3,IS4,IS5,IS6 - SIX STRLINES SELECTED FOR PARTIAL FLOW SOLN.
*                          1) IS1<IS2< (NSR+1) .LE. IS3<IS4<IS5<IS6
*
*  LINE #3
*  XI   - INTEGRATION STEP SIZE (CM)
*  DELX - DELTA-X INCREMENT ALONG SHOCK (CM)
*  ZSTERM - LENGTH OF SYMMETRY AXIS Z (CM)
*  VINFL - FREESTREAM VELOCITY (CM/SEC)
*  PINFL - FREESTREAM PRESSURE (DYNES/CM-2)
*  TINF  - FREESTREAM TEMPERATURE (K)
*
*  LINE #4
*  IS7  - SELECTED STRLNE #. OUTPUT ON T,TVN2, ALL CI'S WITH COORD.
*        PTS X,Z,R,L, AND BODY,SHOCK, AND STRLNE GEOMETRIES (PLOTS)
*        1) IS7 > NSR
*
*  LINE
*  1 : IPF,M,NSR,IANS,MW,MODEL,IQ3,IQ1,IQ2,IQ4,IQ5,IQ7,IQ6,IQ8,IQ9,IQ10
*  2 : IS1,IS2,IS3,IS4,IS5,IS6
*  3 : XI,DELX,ZSTERM,VINFL,PINFL,TINF
*  4 : IS7
*
* -----
50,1,3,0,1,4,1,2,2,6,1,0,0,0,1,0
2,3,6,13,18,20
1.220703125E-04,0.50,0.200,8.915E+05,15.715,197.101
6

```

STATEMENT LISTING OF PROGRAM AFETE.FOR

PROGRAM AFE2

AN INVERSE HYPERSONIC FLOW SOLUTION FOR AN AFE/AOTV BODY IN
CHEMICAL NONEQUILIBRIUM USING VIBRATION-DISSOCIATION COUPLING MODELS,
SHOCK JUMP APPROXIMATIONS, ELECTRON TEMPERATURE MODELING, AND
UNCOUPLED RADIANT HEAT TRANSFER USING A METHOD DERIVED BY S.MASLEN

BASED ON AN ORIGINAL PROGRAM BY B.L. WEIGEL FOR WILLIAM L. GROSE
VIRGINIA POLYTECHNIC INSTITUTE - 1969, (NASA TN D-6529, DEC. 1971)

PROGRAM MODIFICATION BY:

TEXAS A&M UNIVERSITY
DEPARTMENT OF AEROSPACE ENGINEERING
TAMRF-5671 (1987-1988)

PRINCIPAL INVESTAGATOR :	DR. LELAND A. CARLSON
RESEARCH ASSISTANT :	GLENN J. BOBSKILL
RESEARCH ASSISTANT :	ROBERT B. GREENDYKE

CALINTH : INTEGRATION ROUTINE BY CHARLES E. TREANOR
THE NUMERICAL INTEGRATION ALGORITHM USED IS FOUND IN A METHOD
FOR THE NUMERICAL INTEGRATION OF COUPLED FIRST ORDER
DIFFERENTIAL EQUATIONS WITH GREATLY DIFFERENT TIME CONSTANTS

FOFE - EVALUATE E BY NEWTON ITERATION METHOD
FOFTS1 - EVALUATE TS BY NEWTON ITERATION METHOD, CHEMISTRY FROZEN
FOFTS23 - FOFTS1 WITH N2 FROZEN, O2 DISSOCIATING OR BOTH N2 & O2 DISS.
FTLUP - INTERPOLATION ROUTINE

DATA FILES

UNIT 7 IS USED FOR THE INPUT DATA FILE
UNIT 8 IS USED FOR THE OUTPUT DATA FILE
UNIT 9 IS USED TO STORE VIBRATIONAL ENERGY DATA AT THE SHOCK (FORMATTED)
UNIT 10 IS USED TO STORE ALL STAGNATION QUANTITIES
UNIT 11 IS USED TO STORE ALL EVIS DATA FOR COMPUTATIONAL PURPOSES
UNIT 12 IS USED TO STORE ALL PRESSURE DATA
UNIT 13 IS USED TO STORE ALL DPDX DATA
UNIT 14 IS USED TO STORE QUANTITIES FOR PHYSICAL SPACE CALCULATIONS
UNIT 15 IS USED TO STORE QUANTITIES FOR RAD.FOR
UNIT 16 IS USED TO STORE QUANTITIES FOR RAD.FOR
UNIT 17 IS USED TO STORE QUANTITIES FOR RAD.FOR

INPUT-NAMELIST

DELX - INCREMENT ALONG SHOCK , cm
ZSTERM - LENGTH OF SYMMETRY AXIS , Z , cm
IMAX - MAX. NO. OF I-S SPECIES , LESS THAN OR EQUAL TO 25
JMAX - MAX. NO. OF J-S REACTIONS , LESS THAN OR EQUAL TO 50
MJ - CODE INDICATING WHICH SPECIES , I , TO USE TO CALCULATE
COUPLING FACTOR, PHI SUB J , FOR REACTION J
M - 1 FOR VIBRATIONAL NON-EQUILIBRIUM
- 0 FOR VIBRATIONAL EQUILIBRIUM
R - UNIVERSAL GAS CONSTANT , erg/(mole-K)
GAMMA - RATIO OF SPECIFIC HEATS
CIINF - FREESTREAM MASS FRACTION FOR EACH SPECIES

C PINF - FREESTREAM PRESSURE , dynes/cm**2
 C TINF - FREESTREAM TEMPERATURE , K
 C VINP - FREESTREAM VELOCITY , cm/sec
 C MUI - MOLECULAR WT. FOR EACH SPECIES , gm/mole
 C THETAI - CHARACTERISIC VIBRATIONAL TEMPERATURE , K
 C DGENI - FUDGE FACTOR TO PERMIT APPROXIMATING POLYATOMIC MOLECULE
 C BY A DIATOMIC MOLECULE
 C FI - 0 FOR MONATOMIC SPECIES
 C - 1 FOR ALL OTHERS
 C DELHI - HEAT OF FORMATION , ergs/mole
 C DELI - DISSOCIATION ENERGY OF SPECIES , K
 C EVI - VIBRATIONAL ENERGY OF SPECIES , ergs/g
 C BI - $3/2 \ln(2\pi Mi k/h^2) + \ln(k) + FI(I) \ln(THETAI(Rotational))$
 C + $\ln(GIL(I,1))$ FOR EACH SPECIE I
 C LI - NUMBER OF ELECTRONIC LEVELS FOR EACH SPECIES (LI.LE.20)
 C GIL - DEGENERACY OF L-TH ELECTRONIC LEVEL FOR I-TH SPECIES
 C EPSIIL - L-TH ELECTRONIC ENERGY LEVEL FOR I-TH SPECIES , K
 C AJ - FREQUENCY FACTOR IN ARRENIUS TYPE RATE EQN.
 C BJ - TEMPERATURE EXPONENT IN ARRENIUS TYPE EQN.
 C EJ - ACTIVATION ENERGY IN ARRENIUS TYPE EQN.
 C DIRECT - DIRECTION OF THE REACTION (FORW. - 1.0 , BACK. - 2.0)
 C AIJ - FACTOR TO ALLOW USE OF GENERAL SPECIES IN REACTION EQNS.
 C i.e. (N2 + M = 2N + M), AIJ = 1.0 OR (N2 + O2 = 2NO), AIJ = 0.0
 C NUIJ - STOICHIOMETRIC COEFFICIENTS OF I-TH REACTANT IN J-TH REACTION
 C NUPIJ - STOICHIOMETRIC COEFFICIENTS FOR I-TH PRODUCT IN J-TH REACTION
 C ALPIK - FACTORS IN EQN. FOR VIBRATIONAL RELAXATION TIME
 C BETAIK - FACTORS IN EQN. FOR VIBRATIONAL RELAXATION TIME
 C SIGIK - FACTORS IN EQN. FOR VIBRATIONAL RELAXATION TIME
 C NIP - NUMBER OF VIBRATIONAL LEVELS FOR ANHARMONIC OSCILLATOR , K
 C UP - CHARACTERISTIC PROBABILITY TEMPERATURE , K
 C WE - CONST. USED TO COMPUTE ANHARMONIC OSCILLATOR , K
 C WEXE - CONST. USED TO COMPUTE ANHARMONIC OSCILLATOR , K
 C WEYE - CONST. USED TO COMPUTE ANHARMONIC OSCILLATOR , K
 C WEZE - CONST. USED TO COMPUTE ANHARMONIC OSCILLATOR , K
 C XI - INITIAL COMPUTING INTERVAL , .0001220703125 UNLESS INPUT , cm
 C ELE1 - (2*IMAX + 1) VALUES USED BY INTEGRATION SCHEME
 C NORMALLY 0.1,0.5, OR .05 , (UPPER PH RANGE)
 C ELE2 - (2*IMAX + 1) VALUES USED BY INTEGRATION SCHEME
 C NORMALLY .05,.1, OR .01 AND .LT. ELE1 , (LOWER PH RANGE)
 C XPST - 99 OR LESS Xs AT WHICH PHYSICAL SPACE CALCULATIONS ARE
 C DESIRED. THEY MUST BE MULTIPLES OF DELX IN ORDER TO HAVE
 C RS,COST,ZS, AND SINT VALUES AND LAST MUST BE .GT. X AT
 C ZSTERM. THEREFORE, XPST(NXPST) SET = X AT ZSTERM + 100.0
 C IN PROGRAM. XPST(1) MAY NOT BE 0.0 . THEREFORE, SET
 C XPST(1) = DELX IN PROGRAM
 C NXPST - NUMBER OF Xs AT WHICH PHYSICAL SPACE CALCULATIONS ARE
 C DESIRED
 C CIMAX - MAX. CJ OR COMPUTING INTERVAL 0.0625 UNLESS INPUT OTHERWISE
 C HCHCKT - CONTROL ON SIZE OF COMPUTING INTERVAL IN CHECK
 C IF(ABS(HPREV - H)/H.GT.HCHECK) REDUCE INTERVAL
 C TCHCKT - CONTROL ON SIZE OF COMPUTING INTERVAL IN CHECK
 C IF(ABS(TPREV - T)/T.GT.TCHECK) REDUCE INTERVAL
 C PHMAX - CONTROL ON COMPUTING INTERVAL .LE. 65.0
 C IPF - OUTPUT PRINT FREQUENCY
 C
 C STOPS
 C -----
 C STOP 1 INCORRECT INPUT
 C STOP 2 IN MAIN WHEN NXPST IS .LE. 2
 C STOP 13 IN SHOCKG

```

C      STOP 30      IN CHECK WHEN COMPUTING INTERVAL .LT. 1.0E-15
C      STOP 66      IN BASIC WHEN NO CONVERGENCE ON E ITERATION (FOFE)
C      STOP 301     IN MAIN FOR ERROR IN XPST ARRAY OR IZTERM .GT. 200
C      STOP 321     IN MAIN WHEN NO CONVERGENCE ON TS ITERATION (FOFTS)
C      STOP 663     IN MAIN WHEN X .NE. VARI(IPSI)
C      STOP 665     IN MAIN AFTER AN INTEGRATION ATTEMPT
C      STOP 670     IN MAIN WHEN A CI NEGATIVE
C
C      NOTE : PROGRAM IS CURRENTLY CONFIGURED FOR A MAXIMUM
C              200 STREAMLINE SOLUTION
C
C      IMPLICIT REAL*8(A-H,O-Z)
C      COMMON /A1/ M,IX,IMAX,IPSI,MODEL,TE1,TE2
C      COMMON /A2/ P(200),DPDX(200),VARI(200)
C
C      FOLLOWING 7 VARIABLES DIMENSIONED BY IMAX
C      COMMON /A3/ EVIINF(25),THETAI(25),MUI(25),FI(25),DELHI(25),
*      CIINF(25),LI(25)
C      COMMON /A4/ R,MUINF,DELTA,LAMSQ,OMEGSQ,OMEGA,MU,T,U,TINF
C      FOLLOWING 2 VARIABLES DIMENSIONED BY (LMAX IN LI,IMAX)
C      COMMON /A5/ EPSIIL(20,25),GIL(20,25),CID(200,4)
C      COMMON /A6/ VAR(52),CUVAR(52),DER(51)
C      FOLLOWING 4 VARIABLES DIMENSIONED BY JMAX
C      COMMON /A7/ MJ(50),EJ(50),AJ(50),BJ(50),DIRECT(50)
C      FOLLOWING 3 VARIABLES DIMENSIONED BY IMAX
C      COMMON /A8/ TVI(25),NI(25),DGENI(25),BI(25)
C
C      EVIS(IMAX)
C      COMMON /A9/ EVIS(25),XPST(100)
C      FOLLOWING 3 VARIABLES DIMENSIONED BY (IMAX+1,JMAX) OR (IMAX,JMAX)
C      COMMON /A10/ NUIJ(26,50),AIJ(25,50),NUPIJ(25,50)
C      FOLLOWING 3 VARIABLES DIMENSIONED BY (IMAX,JMAX)
C      COMMON /A11/ SIGIK(25,25),ALPIK(25,25),BETAIK(25,25)
C      COMMON /A12/ SINTM(200),COSTM(200),RSM(200),RCM(200),X1(200)
*      ,ZSM(200)
C      COMMON /A13/ DELX,ZSTERM,IZTERM,NSR,MW,SP,TS
C      COMMON /A14/ EINF,PINF,RHOINF,VINF,E,JMAX,KEYINT,RHO,HSTAG
C      COMMON /A15/ PFTL,KITR1,NIP(25),UP(25)
C      COMMON /A16/ IUNEG,WE(25),WEZE(25),WEYE(25),WEZE(25)
C      COMMON /A17/ ELB,SPEC,CJ,TPREV,HPREV,HCHECK,TCHECK
C      COMMON /A18/ ITNEG,IEXP
C      COMMON /A20/ WE1,WE2
C      COMMON /A19/ ELE1(51),ELE2(51),NERR
C
C      DELI(IMAX)
C      DIMENSION DELI(25)
C      DIMENSION RUT(200),PSISTG(200)
C      DIMENSION CIG(25,11),EIG(25,11)
C      DIMENSION PSIG(11),TG(11),HG(11),CIT(11),EIT(11),EG(11)
C
C      CM(IMAX),CI(IMAX),EVI(IMAX)
C      DIMENSION CM(25),CI(25),EVI(25)
C      EQUIVALENCE (VAR(1),XVAR),(VAR(2),H),(VAR(3),CI(1))
C      REAL*8 MUI,MUINF,MU,MINF,MINFSQ,LAMBDA,LAMSQ
C      CHARACTER*30 INFILE,OUTFL1,OUTFL2,OUTFL3
C      CHARACTER*30 SPECIE(25),REACT(50)
C      OPEN GENERAL INPUT FILE CREATED BY INPUT.FOR
C      OPEN (UNIT=106,FILE='GEN.INPUT',STATUS='OLD')
C      READ INTERACTIVE INPUTS FROM GEN.INPUT
C      READ(106,*) IPF,M,NSR,IAN,MW,MODEL,IQ3,IQ1,IQ2,IQ4,
*      IQ5,IQ7,IQ6,IQ8,IQ9
C      READ(106,*) IS1,IS2,IS3,IS4,IS5,IS6
C      READ(106,*) XI,DELX,ZSTERM,VINF,PINF,TINF
C      IF (IQ4.EQ.0) READ(106,20) INFILE,OUTFL1,OUTFL2

```

```

      IF (IQ4.GT.0) READ(106,18) INFILE,OUTFL1,OUTFL2,OUTFL3
18  FORMAT(1X,4(A30,2X))
20  FORMAT(1X,3(A30,2X))
      CLOSE(106)
      NSR1 = NSR + 1
      NERR = 0
      ITNEG = 0
      IEXP = 0
      NEG = 0
      IUNEG = 0
      DO 21 I = 1,52
      VAR(I) = 0.0
      CUVAR(I) = 0.0
      IF (I.EQ.52) GO TO 21
      DER(I) = 0.0
      ELE1(I) = 0.0
      ELE2(I) = 0.0
21  CONTINUE
      OPEN (UNIT=7,FILE=INFILE,STATUS='OLD')
      OPEN (UNIT=8,FILE=OUTFL1,STATUS='UNKNOWN')
      OPEN (UNIT=15,FILE='RADIN1.DAT',STATUS='UNKNOWN')

```

C

READ INPUT

```

      READ(7,*) IMAX,JMAX,R,GAMMA
      READ(7,*) NXPST,CIMAX,PHMAX,TCHCKT,HCHCKT
      NXCON = NXPST - 1
      WRITE(15,40) OUTFL3
40  FORMAT(A30)
      IQ10=1
      WRITE(15,*) MODEL,IQ3,VINF,PINF,TINF
      WRITE(15,*) NXPST,NXCON,IMAX,IQ4,IQ5,IQ6,IQ8,IQ9,IQ10
      DO 42 I = 1,IMAX
      READ(7,*) THETAI(I),MUI(I),FI(I),DELHI(I),CIINF(I),LI(I),DGENI(I)
      *      ,DELI(I),EVI(I),BI(I)
42  WRITE(15,*) MUI(I)
      DO 45 J = 1,JMAX,2
45  READ(7,*) MJ(J),AJ(J),BJ(J),EJ(J),DIRECT(J)
      DO 48 J = 2,JMAX,2
48  READ(7,*) MJ(J),DIRECT(J)
      IMAX21 = 2*IMAX + 1
      DO 51 I = 1,IMAX21
51  READ(7,*) ELE1(I),ELE2(I)
      DO 54 I = 1,IMAX
54  READ(7,*) (AIJ(I,J),J = 1,JMAX)
      DO 57 J = 1,JMAX
57  READ(7,*) (NUIJ(I,J),I = 1,IMAX+1)
      DO 60 J = 1,JMAX
60  READ(7,*) (NUPIJ(I,J),I = 1,IMAX)
      DO 63 I = 1,IMAX
63  READ(7,*) (ALPIK(J,I),J = 1,IMAX)
      DO 66 I = 1,IMAX
66  READ(7,*) (BETAIK(J,I),J = 1,IMAX)
      DO 69 I = 1,IMAX
69  READ(7,*) (SIGIK(J,I),J = 1,IMAX)
      DO 72 I = 1,IMAX
      LII = LI(I)
      WRITE(15,*) LII
      READ(7,*) (GIL(L,I),L = 1,LII)
72  WRITE(15,*) (GIL(L,I),L = 1,LII)
      DO 75 I = 1,IMAX
      LII = LI(I)

```



```

WRITE(15,*) LII
READ(7,*) (EPSIIL(L,I),L = 1,LII)
75 WRITE(15,*) (EPSIIL(L,I),L = 1,LII)
   IF (IANS.EQ.1) THEN
     NXPST = 4
     XPST(1) = (IS3 - NSR)*DELX
     XPST(2) = (IS4 - NSR)*DELX
     XPST(3) = (IS5 - NSR)*DELX
     XPST(4) = (IS6 - NSR)*DELX
     GO TO 78
   ENDIF
   DO 76 I = 1,NXPST
     XPST(I) = DELX*I
76  WRITE(15,*) I,XPST(I)
78  DO 80 I = 1,IMAX
80  READ(7,*) NIP(I),UP(I),WE(I),WEXE(I),WEYE(I),WEZE(I)
     IMAXP2 = IMAX + 2
C                                     SUM OF CIINF SHOULD BE 1.0
     SUM = 0.0
     DO 81 I = 1,IMAX
81  SUM = SUM + CIINF(I)
     IF (SUM.EQ.1.0) GO TO 85
     WRITE(*,82) SUM
82  FORMAT(1X,'SUM OF CIINF(I) SHOULD BE 1: IT IS = ',E11.4,' STOP 1')
     STOP
C                                     STOP 1
85  HCHECK = HCHCKT
     TCHECK = TCHCKT
C                                     READ IN AND WRITE OUT SPECIES AND REACTIONS
     DO 88 I = 1,IMAX
88  READ(7,90) SPECIE(I)
90  FORMAT(A30)
     WRITE(8,95)
     WRITE(*,95)
95  FORMAT(//)
     WRITE(8,100)
     WRITE(*,100)
100 FORMAT(14X,'SPECIES'/)
     DO 110 I = 1,IMAX
105 FORMAT(1X,I5,11X,A30)
     WRITE(*,105) I,SPECIE(I)
110 WRITE(8,105) I,SPECIE(I)
     WRITE(*,95)
     WRITE(8,95)
     DO 112 I = 1,JMAX
112 READ(7,90) REACT(I)
     WRITE(*,115)
     WRITE(8,115)
115 FORMAT(14X,'REACTIONS'/)
     DO 120 I = 1,JMAX
     WRITE(*,125) I,REACT(I)
120 WRITE(8,125) I,REACT(I)
125 FORMAT(1X,I5,8X,A30)
     WRITE(*,95)
     WRITE(8,95)
     CLOSE(7)
C                                     PRINT INPUT
     WRITE(8,135) HCHCKT,TCHCKT,PHMAX
     WRITE(*,135) HCHCKT,TCHCKT,PHMAX
135 FORMAT(1X,'HCHCKT = ',E11.4,4X,'TCHCKT = ',E11.4,4X,'PHMAX = ',

```

```

*      E11.4/)
WRITE(8,140) XI,CIMAX,NXPST
WRITE(*,140) XI,CIMAX,NXPST
140  FORMAT(1X,'XI = ',E11.4,8X,'CIMAX = ',E11.4,5X,'NXPST = ',I3//)
DO 141 I = 1,IMAX21
WRITE(*,142) I,ELE1(I),I,ELE2(I)
141  WRITE(8,142) I,ELE1(I),I,ELE2(I)
142  FORMAT(1X,'ELE1(',I2,') = ',E11.4,5X,'ELE2(',I2,') = ',E11.4)
WRITE(8,95)
WRITE(*,95)
DO 143 I = 1,NXPST
WRITE(*,144) I,XPST(I)
143  WRITE(8,144) I,XPST(I)
144  FORMAT(1X,'XPST(',I3,') = ',E11.4)
WRITE(8,95)
WRITE(*,95)
WRITE(8,145) IMAX,JMAX,M
WRITE(*,145) IMAX,JMAX,M
145  FORMAT(1X,'IMAX = ',I2,11X,'JMAX = ',I2,13X,'M = ',I2//)
WRITE(8,150) DELX,ZSTERM,IPF,R,PINF,TINF,VINF,GAMMA
WRITE(*,150) DELX,ZSTERM,IPF,R,PINF,TINF,VINF,GAMMA
150  FORMAT(1X,'DELX = ',E11.4,2X,'ZSTERM = ',E11.4,2X,
*'PRINT FREQ. = ',I3,3X,'R = ',E11.4,/,1X,'PINF = ',E11.4,
*2X,'TINF = ',E11.4,4X,'VINF = ',E11.4,2X,'GAMMA = ',E9.3//)
WRITE(8,155)
WRITE(*,155)
155  FORMAT(16X,'MUI',8X,'THETAI',7X,'DGENI',10X,'FI',9X,'DELHI'//)
DO 157 I = 1,IMAX
WRITE(*,160) I,MUI(I),THETAI(I),DGENI(I),FI(I),DELHI(I)
157  WRITE(8,160) I,MUI(I),THETAI(I),DGENI(I),FI(I),DELHI(I)
160  FORMAT(1X,I2,8X,5(E11.4,2X))
WRITE(8,95)
WRITE(8,161)
WRITE(*,95)
WRITE(*,161)
161  FORMAT(16X,'DELI',8X,'CIINF',9X,'EVI',10X,'LI',11X,'BI'//)
DO 163 I = 1,IMAX
WRITE(*,164) I,DELI(I),CIINF(I),EVI(I),LI(I),BI(I)
163  WRITE(8,164) I,DELI(I),CIINF(I),EVI(I),LI(I),BI(I)
164  FORMAT(1X,I2,8X,3(E11.4,2X),3X,I3,7X,E11.4)
WRITE(8,95)
WRITE(8,165)
WRITE(*,95)
WRITE(*,165)
165  FORMAT(8X,'NIP',9X,'UP',10X,'WE',10X,'WEXE',9X,'WEYE',8X,'WEZE'//)
DO 167 I = 1,IMAX
166  FORMAT(1X,I2,5X,I3,4X,E10.4,2X,E10.4,3X,E10.4,3X,E10.4,2X,E10.4)
WRITE(*,166) I,NIP(I),UP(I),WE(I),WEXE(I),WEYE(I),WEZE(I)
167  WRITE(8,166) I,NIP(I),UP(I),WE(I),WEXE(I),WEYE(I),WEZE(I)
WRITE(8,95)
WRITE(8,170)
WRITE(*,95)
WRITE(*,170)
170  FORMAT(12X,'MJ',9X,'AJ',14X,'BJ',15X,'EJ',10X,'DIRECTION',/,
*      69X,'F=1,B=2'//)
DO 180 J = 1,JMAX,2
175  FORMAT(1X,I2,2X,I8,3(2X,E15.6),5X,F5.2)
WRITE(*,175) J,MJ(J),AJ(J),BJ(J),EJ(J),DIRECT(J)
180  WRITE(8,175) J,MJ(J),AJ(J),BJ(J),EJ(J),DIRECT(J)
WRITE(8,95)

```

```

        WRITE(8,181)
        WRITE(*,95)
        WRITE(*,181)
181  FORMAT(12X,'MJ',6X,'DIRECTION',/,21X,'F-1,B-2'/)
        DO 183 J = 2,JMAX,2
182  FORMAT(1X,I2,2X,I8,8X,F5.2)
        WRITE(*,182) J,MJ(J),DIRECT(J)
183  WRITE(8,182) J,MJ(J),DIRECT(J)
        WRITE(8,95)
        WRITE(*,95)
        DO 187 I = 1,IMAX
        LII = LI(I)
        DO 186 L = 1,LII
        WRITE(8,185) L,I,GIL(L,I),L,I,EPSIIL(L,I)
        WRITE(*,185) L,I,GIL(L,I),L,I,EPSIIL(L,I)
185  FORMAT(1X,'GIL(',I2,',',I2,') = ',E11.4,4X,
        * 'EPSIIL(',I2,',',I2,') = ',E11.4)
186  CONTINUE
187  CONTINUE
        WRITE(8,95)
        WRITE(*,95)
        DO 200 I = 1,IMAX
        DO 196 J = 1,JMAX
        WRITE(*,195) I,J,AIJ(I,J)
        WRITE(8,195) I,J,AIJ(I,J)
195  FORMAT(1X,'AIJ(',I2,',',I2,') = ',E11.4)
196  CONTINUE
200  CONTINUE
        WRITE(8,95)
        WRITE(*,95)
        DO 205 I = 1,IMAX+1
        DO 202 J = 1,JMAX
        WRITE(8,201) I,J,NUIJ(I,J),I,J,NUPIJ(I,J)
        WRITE(*,201) I,J,NUIJ(I,J),I,J,NUPIJ(I,J)
201  FORMAT(1X,'NUIJ(',I2,',',I2,') = ',I5,9X,'NUPIJ(',I2,',',
        * I2,') = ',I5)
202  CONTINUE
205  CONTINUE
        WRITE(8,95)
        WRITE(*,95)
        DO 225 I = 1,IMAX
        DO 220 J = 1,IMAX
        WRITE(8,210) I,J,SIGIK(I,J),I,J,ALPIK(I,J),I,J,BETAIK(I,J)
        WRITE(*,210) I,J,SIGIK(I,J),I,J,ALPIK(I,J),I,J,BETAIK(I,J)
210  FORMAT(1X,'SIGIK(',I2,',',I2,') = ',E10.4,1X,
        * 'ALPIK(',I2,',',I2,') = ',E10.4,1X,'BETAIK(',I2,',',I2,') = ',
        * E10.4)
220  CONTINUE
225  CONTINUE
        WRITE(8,95)
        WRITE(8,230)
        WRITE(*,95)
        WRITE(*,230)
230  FORMAT(7X,25('*'),' END INPUT ',25('*')///)
        WRITE(8,231)
        WRITE(*,231)
231  FORMAT(25X,'+++ SHOCK GEOMETRY +++'//)
        NXPSTM1 = NXPST - 1

```

C

CALL SHOCKG (NXPSTM1,ITK)

DEFINE SHOCK GEOMETRY

```

C          IZTERM = NO. OF DELTA X INCREMENTS GENERATED IN SHOCKG
IF (IQ4.GT.0.AND.IZTERM.GT.30) THEN
  WRITE(*,232) IZTERM
232  FORMAT(1X,'IZTERM = ',I4,2X,'WITH A RADIATION MODEL SELECTED'
*      ,/,1X,'MAXIMUM NO. OF STREAMLINES .LE. 30')
  STOP
ENDIF
WRITE(8,95)
WRITE(*,95)
IF ((NXPST - 1.0).LE.IZTERM) GO TO 237
WRITE(*,235)
235  FORMAT(1X,'(NXPST-1).GT.IZTERM : IT MUST BE .LE. TO IZTERM',/,
*      1X,'PHYSICAL SPACE CALCULATIONS - SEE DO 700 LOOP : STOP 301'/)
  STOP
237  WRITE(8,95)
  WRITE(*,95)
  IF (IZTERM.LE.200) GO TO 241
  WRITE(*,240)
240  FORMAT(1X,'IZTERM.GT.200:CHGE DIM OF P,DPDX AND VARI: STOP 301'/)
  STOP

C          FREESTREAM QUANTITIES
241  SUM = 0.0
  DO 245 I = 1,IMAX
245  SUM = SUM + CIINF(I)/MUI(I)
  MUINF = 1.0/SUM
  RHOINF = MUINF*PINF/(R*TINF)
  AINF = DSQRT(GAMMA*R*TINF/MUINF)
  MINF = VINP/AINF

C          FOR EACH I , SPECIE
  EINF = 0.0
  DO 275 I = 1,IMAX
  TEM = DEXP(THETAI(I)/TINF)
  EVIINF(I) = (R*THETAI(I))/(MUI(I)*(TEM - 1.0))*FI(I)

C          FOR EACH L, REACTION LEVEL
  GSUM = 0.0
  GESUM = 0.0
  LII = LI(I)
  IF (LII.LE.20) GO TO 255
  WRITE(*,250) LII
250  FORMAT(1X,'LII = ',I3,2X,'A LEVEL IN LI ARRAY IS GREATER THAN 20',
*      2X,/,1X,'YOU NEED TO CHANGE DIMENSION OF EPSIIL AND GIL')
255  CONTINUE
  DO 265 L = 1,LII
  TEM1 = DEXP(-EPSIIL(L,I)/TINF)
  GSUM = GSUM + GIL(L,I)*TEM1
  GESUM = GESUM + GIL(L,I)*EPSIIL(L,I)*TEM1
265  CONTINUE
  EEIINF = R/MUI(I)*(GESUM/GSUM)
  EIINF = 1.5*R*TINF/MUI(I) + FI(I)*R*TINF/MUI(I) + EVIINF(I)
*      + EEIINF + DELHI(I)/MUI(I)
  EINF = EINF + EIINF*CIINF(I)
275  CONTINUE
  WRITE(8,280) MUINF,RHOINF,AINF,TINF,PINF,VINF,MINF,EIINF,EINF,
*      EEIINF
  WRITE(*,280) MUINF,RHOINF,AINF,TINF,PINF,VINF,MINF,EIINF,EINF,
*      EEIINF
280  FORMAT(24X,'+++ FREESTREAM QUANTITIES +++',///,6X,'MUINF = ',E11.4
*,3X,'RHOINF = ',E11.4,3X,'AINF = ',E11.4,///,6X,'TINF = ',E11.4,3X,
*'PINF = ',E11.4,3X,'VINF = ',E11.4,///,1X,'MINF = ',E10.4,2X,
*'EIINF = ',E10.4,2X,'EINF = ',E10.4,2X,'EEIINF = ',E10.4//)

```

```

DO 285 I = 1,IMAX
WRITE(*,286) I,EVIINF(I)
285 WRITE(8,286) I,EVIINF(I)
286 FORMAT(1X,'EVIINF(',I2,') = ',E11.4)
WRITE(8,95)
WRITE(*,95)

C BEGIN QUANTITIES BEHIND SHOCK FOR THE RANGE OF X

WRITE(8,290)
WRITE(*,290)
290 FORMAT(22X,'+++ QUANTITIES BEHIND SHOCK +++',///,5X,'TS',9X,'ES',
*9X,'PS',7X,'RHOS',8X,'US',8X,'PSIS',8X,'HS'/)
IF (NXPST.GE.3) GO TO 293
WRITE(*,291)
STOP
291 FORMAT(1X,'NXPST = 2 - SURELY SOME PHYSICAL SPACE VALUES',/,1X,
* 'ARE DESIRED -- MAKE IT AT LEAST 3 : STOP 2')
293 OPEN (UNIT=9,FILE=OUTFL2,STATUS='UNKNOWN')
OPEN (UNIT=10,FILE='STAG.DAT',STATUS='UNKNOWN')
OPEN (UNIT=11,FILE='EVISC.DAT',STATUS='UNKNOWN')

C
DO 385 IX = 1,IZTERM
TEM = SINTM(IX)**2
LAMBDA = RHOINF*VIN*F*SINTM(IX)
OMEGA = PINF + RHOINF*VIN*F**2*TEM
DELTA = EINF + PINF/RHOINF + (VIN*F**2*TEM)/2.0
LAMSQ = LAMBDA**2
OMEGSQ = OMEGA**2
MINFSQ = MINF**2
TEM = TEM*MINFSQ
TSG = (TINF*(2.0*GAMMA*TEM - (GAMMA - 1.0))*((GAMMA - 1.0)
* TEM + 2.0))/((GAMMA + 1.0)**2*TEM)
TOL1 = .001
TOL2 = .0001
MAXI = 50

C
IF (IQ3.EQ.1) CALL FOFTS1 (MAXI,TOL1,TOL2,ICODE,TSG)
IF (IQ3.GT.1) CALL FOFTS23 (MAXI,TOL1,TOL2,ICODE,TSG,IQ1,IQ2,IQ3)

C
IF (ICODE.EQ.1) WRITE(*,330) TSG
330 FORMAT(1X,'**** MAXIMUM ITERATION EXCEEDED IN FOFTS ****',//,
* 1X,'LAST ITERATED VALUE OF TSG WAS ',E15.6,' : STOP 321'/)
IF (ICODE.EQ.2) WRITE(*,335)
335 FORMAT(1X,'***** DERIVATIVE = 0.0 IN FOFTS ***** : STOP 321'/)
IF (ICODE.EQ.1.OR.ICODE.EQ.2) STOP

C STOP 321
IF (IX.EQ.1) WRITE(15,*) TSG
TS = TSG
ES = 0.0
DO 360 I = 1,IMAX
IF (M.EQ.1) EVIS(I) = EVIINF(I)
IF (M.EQ.1) GO TO 345
TEM = DEXP(THETAI(I)/TS)
EVIS(I) = (R*THETAI(I))/(MUI(I)*(TEM - 1.0))*FI(I)
345 SUMG = 0.0
SUMGE = 0.0
LII = LI(I)
DO 350 L = 1,LII
TEM1 = DEXP(-EPSIIL(L,I)/TS)
SUMG = SUMG + TEM1*GIL(L,I)
350 SUMGE = SUMGE + TEM1*GIL(L,I)*EPSIIL(L,I)

```

```

      EEIS = (R/MUI(I))*(SUMGE/SUMG)
      EIS = (1.5*R*TS)/MUI(I) + (FI(I)*R*TS)/MUI(I) + EVIS(I)
      *      + EEIS + DELHI(I)/MUI(I)
      IF (IQ3.EQ.1) THEN
        ES = ES + EIS*CIINF(I)
      ELSE
        ES = ES + EIS*CID(IX,I)
      ENDIF
360  CONTINUE
      PS = DSQRT(OMEGSQ - 2.0*LAMSQ*(DELTA - ES))
      RHOS = (MUINF*PS)/(R*TS)
      US = VINFINF*COSTM(IX)
      PSIS = (RHOINF*VINFINF*RSM(IX)**2)/2.0
      HS = PS/RHOS + ES
      WRITE(10,*) IX,TS,ES,PS,RHOS,US,PSIS,HS
      WRITE(8,365) TS,ES,PS,RHOS,US,PSIS,HS
      WRITE(*,365) TS,ES,PS,RHOS,US,PSIS,HS
365  FORMAT(1X,7(E9.4,2X))
      WRITE(9,366) IX
366  FORMAT(1X,'FOR IZTERM = ',I3//)
      DO 370 I = 1,IMAX
        WRITE(9,367) I,EVIS(I)
367  FORMAT(13X,'EVIS(',I2,') = ',E11.4)
370  CONTINUE
        WRITE(11,*) (EVIS(I),I = 1,IMAX)
        WRITE(9,620)
385  CONTINUE
C
      CLOSE(9)
      CLOSE(10)
      CLOSE(11)
      OPEN (UNIT=10,FILE='STAG.DAT',STATUS='OLD')
      OPEN (UNIT=11,FILE='EVISC.DAT',STATUS='OLD')
C
      COMPUTE PRESSURE DISTRIBUTION FOR EACH PSI
      OPEN (UNIT=12,FILE='PRESS.DAT',STATUS='UNKNOWN')
      DO 390 IPSI = 1,IZTERM
      DO 388 IX = 1,IZTERM
        IF (IX.EQ.IPSI) XX = X1(IX)
        READ(10,*) IXT,TS,ES,PS,RHOS,US,PSIS,HS
        IF (IXT.LT.IPSI) GO TO 388
        IF (IXT.EQ.IPSI) PSISHK = PSIS
        IF (IPSI.EQ.IX) THEN
          P(IX) = PS
          GO TO 388
        ENDIF
        P(IX) = PS + US/(RCM(IX)*RSM(IX))*(PSISHK - PSIS)
388  CONTINUE
      CLOSE(10)
      OPEN (UNIT=10,FILE='STAG.DAT',STATUS='OLD')
      WRITE(12,*) XX,(P(IX),IX = 1,IZTERM),IPSI,IZTERM
390  CONTINUE
      CLOSE(12)
      OPEN (UNIT=12,FILE='PRESS.DAT',STATUS='OLD')
      OPEN (UNIT=13,FILE='DPDX.DAT',STATUS='UNKNOWN')
C
      COMPUTE DPDX AND VARI
      DO 404 IPSI = 1,IZTERM
      READ(12,*) X,(P(I),I=1,IZTERM),IPI,IZTERM
      ITHIS = 0
      DO 401 IX = IPSI,IZTERM
        IF (IX.EQ.IPSI) XT = X

```

```

      IF (IX.GT.IPSI.AND.IPSI.LT.(NSR+2)) XT = XT + DELX/NSR
      IF (IX.GT.IPSI.AND.IPSI.GT.(NSR+1)) XT = XT + DELX
      ITHIS = ITHIS + 1
      VARI(IThis) = XT
401  CONTINUE
      DO 403 IX = IPSI,IZTERM
C
      DIFFERENTIATE
      IF (IX.NE.IPSI) GO TO 402
      DPDX(IX) = (- 3.*P(IX) + 4.*P(IX+1) - P(IX+2))/(2.0*DELX)
      GO TO 403
402  DPDX(IX) = (- P(IX-1) + P(IX+1))/(2.0*DELX)
      IF (IX.NE.IZTERM) GO TO 403
      DPDX(IX) = (P(IX-2) - 4.*P(IX-1) + 3.*P(IX))/(2.0*DELX)
403  CONTINUE
      WRITE(13,*) (DPDX(I),I=1,IZTERM),IPSI,IZTERM
404  CONTINUE
      CLOSE(13)
      CLOSE(12)
      OPEN (UNIT=12,FILE='PRESS.DAT',STATUS='OLD')
      OPEN (UNIT=13,FILE='DPDX.DAT',STATUS='OLD')
C
      DO 405 IX = 1,IZTERM
      IF (IX.EQ.1) VARI(1) = 0.0
      IF (IX.EQ.1) GO TO 405
      IF (IX.LT.(NSR+2)) VARI(IX) = VARI(IX-1) + DELX/NSR
      IF (IX.GT.(NSR+1)) VARI(IX) = VARI(IX-1) + DELX
405  CONTINUE
      WRITE(8,95)
      WRITE(*,95)
      WRITE(8,407) ITK,NXPST
      WRITE(*,407) ITK,NXPST
407  FORMAT(1X,'XXXXX ITK = ',I3,5X,'NXPST = ',I3,' XXXXX'//)
      IF (ITK.GE.(2*NXPST/4)) GO TO 409
      WRITE(*,408)
408  FORMAT(1X,'NO. OF X-S IN XPST ARRAY,ITK, IS .LT. .5*NXPST',/
*      ,1X,'REEXAMINE DELX,NXPST AND XPST ARRAY : STOP 301'/)
      STOP
C
      STOP 301
C
C  END COMPUTATION OF QUANTITIES BEHIND SHOCK ; INTEGRATE ALONG EACH PSI
C
C  VAR(1) = XVAR
C  VAR(2) = H,ENTHALPY          VAR(2) MUST BE H, WHICH MAY BE + OR -
C
C                                CONCENTRATION OF SPECIE
C  VAR(3)      = CI(1)
C  VAR(2+IMAX) = CI(IMAX)
C
C                                EQUILIBRIUM VIBRATIONAL ENERGY
C  VAR(2+IMAX+1) = EVI(1)
C  VAR(2+IMAX+IMAX) = EVI(IMAX)
C
409  WRITE(8,411)
      WRITE(*,411)
411  FORMAT(1X,'NO. OF VIB. LEVELS TO DISSOCIATION',/)
      DO 410 I = 1,IMAX
C
      TRUNCATE
      YIY = FI(I)*(DELI(I)/THETAI(I) + 1.0)
      NI(I) = AINT(YIY)
      WRITE(*,412) I,NI(I)
410  WRITE(8,412) I,NI(I)

```

```

412  FORMAT(11X,'NI(',I2,') - ',I3)
      WRITE(*,95)
      WRITE(8,95)
      MU = 0.0
      N = 1 + 2*IMAX
      IF (M.EQ.0) N = N - IMAX
C
                                     INITIALIZE
      WRITE(8,415) IZTERM
      WRITE(*,415) IZTERM
415  FORMAT(1X,'IZTERM = ',I4///)
      WRITE(8,417)
      WRITE(*,417)
417  FORMAT(1X,'COUPLED VIBRATION-DISSOCIATION MODELS',//,12X,
* 'TYPE',11X,'NO.',//,9X,'VIB. EQUIL.',8X,'0',/,12X,'CVD',13X,
* '1',/,12X,'CVDV',12X,'2',/,6X,'CVDV-Preferential',5X,'3',/,
* 12X,'PARK',12X,'4'///)
      WRITE(8,418)
      WRITE(*,418)
418  FORMAT(7X,'SHOCK JUMP CONDITION MODELS',//,12X,
* 'TYPE',11X,'NO.',//,6X,'CHEMISTRY FROZEN',6X,'1',/,5X,
* 'N2 FROZEN, O2 DISS.',4X,'2',/,7X,'N2 AND O2 DISS.',6X,'3'///)
      IF (MW.EQ.1) THEN
        WRITE(*,419)
        WRITE(8,419)
419  FORMAT(3X,'MILLIKAN AND WHITE DATA FOR N2 WAS SELECTED'/)
      ENDIF
      KSTAG = 0
      ISTAG = 0
      XPST(1) = DELX
      XPST(NXPST) = VARI(IZTERM) + 100.0
      SUMCI = 0.0
      WRITE(*,95)
      WRITE(8,95)
      LPS1 = 1
      IPSI = 0
      OPEN (UNIT=14,FILE='RHORT.DAT',STATUS='UNKNOWN')
      OPEN (UNIT=16,FILE='RADIN2.DAT',STATUS='UNKNOWN')
C
                                     BEGIN EACH STREAMLINE COMPUTATION HERE
420  IPSI = IPSI + 1
      IF (IPSI.EQ.IZTERM) GO TO 735
425  KIPF = 0
C
                                     ISTAG = 2  FOR STREAMLINE DELX
      IF (ISTAG.EQ.1) ISTAG = 2
      READ(10,*) IX,TS,ES,PS,RHOS,US,PSIS,HS
      READ(11,*) (EVIS(I),I = 1,IMAX)
      IF (IPSI.EQ.1) HSTAG = HS
      CJ = XI
      SPEC = 0.0
C
                                     EVALUATE DERIVATIVES WHEN SPEC = 0.0
      II = 0
C
                                     USE SHOCK VALUES FOR INITIAL COMPUTATION ON EACH STREAMLINE
      KEYINT = 0
      IX = 0
      T = TS
      IF (M.EQ.0) GO TO 450
      DO 445 I = 1,IMAX
445  TVI(I) = TINF
      TEL=TINF
      GO TO 460
450  DO 455 I = 1,IMAX

```



```

455 TVI(I) = TS
    TEL=TS
460 RHO = RHOS
    IF (IQ3.GT.1) THEN
        SUM = 0.0
        DO 462 I = 1,IMAX
462  SUM = SUM + CID(IPSI,I)/MUI(I)
        MU = 1.0/SUM
    ENDIF
    IF (IQ3.EQ.1) MU = MUINF
    U = US
    DO 465 I = 1,IMAX
        MM = 2 + I
        K = IMAX + MM
        VAR(MM) = 0.0
        IF (IQ3.EQ.1) VAR(MM) = CIINF(I)
        IF (IQ3.GT.1) VAR(MM) = CID(IPSI,I)
        VAR(K) = 0.0
        IF (M.EQ.0) VAR(K) = EVIS(I)
        IF (M.EQ.1) VAR(K) = EVIINF(I)
        EVI(I) = VAR(K)
465  CONTINUE
C      COMPUTE S EXPONENT FOR PARK MODEL
    IF (MODEL.EQ.4) THEN
        SP = 3.5*DEXP(-5000.0/TS)
    ENDIF
    E = ES
    H = HS
    IF ((IPSI.EQ.(NSR+1)).AND.(ISTAG.EQ.0)) GO TO 635
    READ(12,*) X,(P(L),L=1,IZTERM),IPSI,IZTERM
    READ(13,*) (DPDX(L),L=1,IZTERM),IPSI,IZTERM
    VAR(1) = X
C      OMIT PSI = 0.0 STREAMLINE FOR NOW - PICK IT UP LATER
    IF (X.NE.0.0) GO TO 490
    IG = 0
    WRITE(*,725)
    WRITE(8,725)
    GO TO 730
490 IF (IANS.EQ.1) THEN
    IF ((IPSI.EQ.IS1).OR.(IPSI.EQ.IS2).OR.(IPSI.EQ.IS3)) GO TO 492
    IF ((IPSI.EQ.IS4).OR.(IPSI.EQ.IS5).OR.(IPSI.EQ.IS6)) GO TO 492
    GO TO 730
ENDIF
492 IF (DABS(X - VARI(IPSI)).LE.1.0E-06) GO TO 500
    WRITE(*,495) IPSI,X,VARI(IPSI)
495 FORMAT(1X,'IPSI = ',I3,2X,'X = ',E10.5,2X,'VARI(IPSI) = ',E10.5,
    *//,1X,'X AND VARI(IPSI) SHOULD BE EQUAL : EXAMINE GENERATION OF X
    * AND VARI IN MAIN : STOP 663'//)
    STOP
C      STOP 663
C      INITIALIZE
500 IF (IANS.EQ.1) THEN
    IF ((IPSI.EQ.IS1).OR.(IPSI.EQ.IS2).OR.(IPSI.EQ.IS3)) GO TO 501
    IF ((IPSI.EQ.IS4).OR.(IPSI.EQ.IS5).OR.(IPSI.EQ.IS6)) GO TO 501
    GO TO 730
ENDIF
501 ELB = 0.0
    SPEC = 0.0
    IF (VAR(1).GT.XPST(LPS1)) LPS1 = LPS1 + 1
    LPS2 = LPS1

```

```

505 IX = IX + 1
C
CALL CALINTH (N,CIMAX,PHMAX)
C
IF (NERR.EQ.0) GO TO 575
IF (NERR.EQ.1) WRITE(*,565) CJ,N
565 FORMAT(1X,'BAD INPUT IN CALINTH',/,1X,'CJ = ',E10.5,2X,'N = ',I4
*      ,/,1X,'STOP 665'/)
IF (NERR.EQ.2) WRITE(*,570)
570 FORMAT(1X,'EXAMINE ELE1 AND ELE2 IN CALINTH : STOP 665'/)
IF ((NERR.EQ.1).OR.(NERR.EQ.2)) STOP
C
STOP 665
575 IF (IX.EQ.1) THEN
IF (IQ3.NE.1) THEN
IF (IQ2.EQ.1) TE = TINF
IF (IQ2.EQ.2) TE = TS
TE1 = TE
TE2 = TE
ELSE
TE = TINF
TE1 = TE
TE2 = TE
ENDIF
ENDIF
C
ELECTRON TEMPERATURE MODEL
IF (IX.EQ.1) GO TO 576
C
CORR = 4.23E-06*(T**(-2.88))
CN = RHO*6.02252E+23
C
CNN = CN*VAR(5)/MUI(3)
C
CNO = CN*VAR(6)/MUI(4)
C
TE = DLOG(1. + 10.1/((CNO+CNN)*CORR))
C
TE = T/(1. + TE*T/85000.)
ITRTE=0
FNA=CN*VAR(5)/MUI(3)
FNI=CN*VAR(11)/MUI(9)
FNE=CN*VAR(9)/MUI(7)
FNM=CN*VAR(3)/MUI(1)
FOA=CN*VAR(6)/MUI(4)
FOI=CN*VAR(12)/MUI(10)
FOM=CN*VAR(4)/MUI(2)
EI=2.3322D-11
EO=2.1735D-11
FAV=6.023D23
SA=1.52D-15
F1=0
F2=0
WE2=WE2*U*FAV/VAR(9)
WE1=WE1*U*FAV/VAR(9)
FO=0
574 ITRTE=ITRTE+1
IF(ITRTE.GT.1)GO TO 571
KK=1
569 KK=KK+1
IF(KK.GT.50)TE1=2000
IF(KK.GT.50)GO TO 576
TE=1000*KK
571 SM=0.5355D-19*TE+.696D-15
IF(ITRTE.GT.50)GO TO 573
SI=1.53614D08*(TE**3)/FNE
SI=DSQRT(SI)

```

```

SI=(4.38384D-06/(TE**2))*DLOG(SI)
FKF=4.16D13*(TE**( 0.50D0))*DEXP(-1.20D05/TE)
FK1=5.49D13*(TE**( 0.50D0))*DEXP(-1.045D05/TE)
FKB=2.3D21*(TE**(-1.0D00))*DEXP(4.9D4/TE)
FK5=1.53D22*(TE**(-1.0D00))*DEXP(5.3D4/TE)
WEA=FKF*FNA-FKB*(FNI/FAV)*FNE
W2E=FK1*FOA-FK5*(FOI/FAV)*FNE
SX=(FNA+FOA)*SA+(FNI+FOI)*SI+(FNM+FOM)/2.0D0*SM
C WEA=WE1
C W2E=WE2
W2=DER(8)*U*FAV/VAR(9)
W1=WEA+W2E
W1=W2
C WRITE(*,563)KK,ITRTE,WEA,W2E,W1
563 FORMAT(' MAIN',2I5,3E11.3)
FE=T-1.23357D-10/SX/DSQRT(TE)*(WEA*EI+W2E*EO+W1*3.45D-16*TE)-TE
IF(ITRTE.GT.1)GO TO 568
IF((FE/DABS(FE)*F0).LT.0.D00)GO TO 567
F0=FE
T0=TE
GO TO 569
567 F1=FE
T1=TE
566 TE=(T0*F1-T1*F0)/(F1-F0)
IF (DABS(TE-T1).LT.1.D00)GO TO 573
IF(DABS(TE-T0).LT.1.D00)GO TO 573
GO TO 574
568 IF((FE*F0).LT.0.0D00)GO TO 564
T0=TE
F0=FE
GO TO 566
564 F1=FE
T1=TE
GO TO 566
C 573 CONTINUE
573 IF (TE.GT.T)TE=T
TE1 = TE
IF (ITRTE.GT.50)WRITE(*,572)
572 FORMAT(' ','TE DID NOT CONVERGE')
CN2 = RHO*6.02252E+23*VAR(3)/MUI(1)
A = DLOG(1.01D+01) - DLOG(CN2) - DLOG(6.4D+00)
* - 59.*DLOG(1.0D+01) + 19.*DLOG(TVI(1))
A = DEXP(A)
TE = DLOG(1. + A)
TE = TVI(1)/(1. + TE*TVI(1)/85000.)
TE2 = TE
C
576 IF (IX.EQ.1) GO TO 580
IF ((IPSI.EQ.(NSR+1)).AND.(IX.EQ.(NSR+1))) GO TO 580
IF (VAR(1).GE.VARI(IZTERM)) GO TO 580
IF ((IPSI.LT.(NSR+1)).AND.(VAR(1).GE.DELX)) GO TO 580
KIPF = KIPF + 1
IF (KIPF.NE.IPF) GO TO 625
KIPF = 0
580 SUMCI = 0.0
DO 590 ISUM = 3,IMAXP2
IF (VAR(ISUM).LT.0.0) NEG = -1
IF (VAR(ISUM).LT.0.0) WRITE(*,585)
585 FORMAT(5X,'----- NEGATIVE CI -----'/)
590 SUMCI = SUMCI + VAR(ISUM)

```

```

DO 595 ICM = 1, IMAX
595 CM(ICM) = VAR(ICM+2)*MU/MUI(ICM)
C
WRITE(8,600) IPSI,IX,IQ3,MODEL,XVAR,H,MU,PFTL,RHO,U,T,E,SPEC,SUMCI
WRITE(*,600) IPSI,IX,IQ3,MODEL,XVAR,H,MU,PFTL,RHO,U,T,E,SPEC,SUMCI
600 FORMAT(1X,'PSI = ',I4,3X,'IX = ',I4,8X,'SHOCK J COND. = ',I2,14X,
*'CVD MODEL = ',I2,/ ,1X,'X = ',D9.4,2X,'H = ',D9.4,2X,'MU = ',
'E9.4,2X,'P = ',E9.4,3X,'RHO = ',E9.4,/ ,1X,'U = ',E9.4,2X,'T = ',
'E9.4,2X,'E = ',E9.4,2X,'CIT = ',E9.4,2X,'SUMCI = ',E9.4 )
DO 603 I = 3, IMAXP2
J = I - 2
WRITE(*,605) J,VAR(I),J,CM(J),J,VAR(IMAX+I)
603 WRITE(8,605) J,VAR(I),J,CM(J),J,VAR(IMAX+I)
605 FORMAT(6X,'CI(',I2,') = ',D11.4,4X,'CM(',I2,') = ',E11.4,4X,
* 'EVI(',I2,') = ',D11.4)
C
WRITE(*,620)
WRITE(8,620)
620 FORMAT(/)
DO 621 I = 1, IMAX
IF (EVI(I).EQ.0.0) TVI(I) = 0.0
IF (I.EQ.IMAX) THEN
WRITE(*,623) I,TVI(I),TE1,ITRTE,W1
WRITE(8,623) I,TVI(I),TE1,ITRTE,W1
GO TO 621
ENDIF
WRITE(*,622) I,TVI(I)
WRITE(8,622) I,TVI(I)
621 CONTINUE
622 FORMAT(5X,'TVI(',I2,') = ',E11.4)
623 FORMAT(5X,'TVI(',I2,') = ',E11.4,1X,'TE1= ',E11.4,1X,
* 'ITRTE= ',I2,1X,'W1= ',E10.3)
C
WRITE(*,620)
WRITE(8,620)
625 IF (NEG.EQ.-1) WRITE(*,630)
630 FORMAT(1X,' A NEGATIVE CONCENTRATION , CI IN MAIN : STOP 670'//)
IF (NEG.EQ.-1) STOP
C
C WHEN IPSI = (NSR+1), DELX AND STAGNATION STREAMLINES WILL BE COMPUTED
C
C IUNEG = 1 IF U**2 NEG. IN BASIC
C
IF (IUNEG.EQ.1) GO TO 735
IF ((IPSI.LT.(NSR+1)).AND.(VAR(1).LT.XPST(1))) GO TO 505
IF ((ISTAG.EQ.1).OR.(ISTAG.EQ.2)) GO TO 690
C
ISTAG = 0 UNTIL AFTER EXTRAPOLATION FOR PSI = 0.0 STREAMLINE
C
ISTAG = 1 FOR PSI = 0.0 STREAMLINE
C
ISTAG = 2 FOR DELX STREAMLINE AND THEREAFTER
635 IG = IG+1
IGC = 0
IGE = IMAX + 2
PSIG(IG) = PSIS
DO 640 IG1 = 3, IMAXP2
IGC = IGC + 1
CIG(IGC,IG) = VAR(IG1)
IGE = IGE + 1
EIG(IGC,IG) = VAR(IGE)
640 CONTINUE
EG(IG) = E
TG(IG) = T
HG(IG) = H
IF (IPSI.LT.(NSR+1)) WRITE(8,725)
IF (IPSI.LT.(NSR+1)) WRITE(*,725)

```

```

      IF (IPSI.LT.(NSR+1)) GO TO 730
C      IPSI = (NSR+1) IS THE STAGNATION STREAMLINE ;
C      AT DELX ,EXTRAPOLATE FOR PSI = 0.0 VALUES

      PSI = 0.0
      MG = 1
      CALL FTLUP (PSI,E,MG,IG,PSIG,EG)
      CALL FTLUP (PSI,T,MG,IG,PSIG,TG)
      CALL FTLUP (PSI,H,MG,IG,PSIG,HG)
      DO 650 IG2 = 1,IMAX
      DO 645 IG1 = 1,NSR1
      CIT(IG1) = CIG(IG2,IG1)
645 EIT(IG1) = EIG(IG2,IG1)
      CALL FTLUP (PSI,CI(IG2),MG,IG,PSIG,CIT)
      IF (CI(IG2).LT.0.0) CI(IG2) = 1.0E-08
      CALL FTLUP (PSI,EVI(IG2),MG,IG,PSIG,EIT)
650 VAR(2+IMAX+IG2) = EVI(IG2)

C      INITIALIZE FOR PSI = 0.0 STREAMLINE

      CJ = XI
      SPEC = 0.0
      II = 0
      KEYINT = 0
      IX = 0
      SUM = 0.0
      DO 660 I = 1,IMAX
      SUM = SUM + CI(I)/MUI(I)
      IF (EVI(I).EQ.0.0) TVI(I) = 0.0
      IF (EVI(I).EQ.0.0) GO TO 660
      XL = ((DGENI(I)*R*THETAI(I))/(MUI(I)*EVI(I)) + 1.0)
      ALN = DLOG(XL)
      TVI(I) = THETAI(I)/ALN
660 CONTINUE
      MU = 1.0/SUM
      U = DSQRT(2.0*(HSTAG - H))
      VAR(1) = DELX

C      COMPUTE P AND DPDX AT IPSI = (NSR+1) ( AT THE BODY )
      IF (IANS.EQ.0) WRITE(8,665)
      IF ((IPSI.EQ.IS1).OR.(IPSI.EQ.IS2).OR.(IPSI.EQ.IS3)) WRITE(8,665)
      IF ((IPSI.EQ.IS4).OR.(IPSI.EQ.IS5).OR.(IPSI.EQ.IS6)) WRITE(8,665)
      IF (IANS.EQ.0) WRITE(*,665)
      IF ((IPSI.EQ.IS1).OR.(IPSI.EQ.IS2).OR.(IPSI.EQ.IS3)) WRITE(*,665)
      IF ((IPSI.EQ.IS4).OR.(IPSI.EQ.IS5).OR.(IPSI.EQ.IS6)) WRITE(*,665)
665 FORMAT(1X,'-- PSI = 0.000 STREAMLINE --'/)
      P(NSR+1) = PS + US/(RCM(NSR+1)*RSM(NSR+1))*(-PSIS)
      RHO = P(NSR+1)/(H - E)
      NSR2 = NSR + 2
      DO 670 I = NSR2,IZTERM
      READ(10,*) IXT,TS,ES,PS,RHOS,US,PSIS,HS
670 P(I) = PS + US/(RCM(I)*RSM(I))*(-PSIS)
      CLOSE(10)
      CLOSE(11)
      DPDX(NSR+1) = (-3.*P(NSR1)+4.*P(NSR2)-P(NSR3))/(2.*DELX)

C      EVALUATE DPDX FROM (NSR+2) TO IZTERM

      DO 680 I = NSR2,IZTERM
      IF (I.LT.IZTERM) THEN
      DPDX(I) = (- P(I-1) + P(I+1))/(2.*DELX)
      ELSE
      DPDX(I) = (P(I-2) - 4.*P(I-1) + 3.*P(I))/(2.*DELX)
      ENDIF
680 CONTINUE
      OPEN (UNIT=10,FILE='STAG.DAT',STATUS='OLD')

```

```

OPEN (UNIT=11,FILE='EVISC.DAT',STATUS='OLD')
DO 685 I = 1,NSR
READ(10,*) IX,TS,ES,PS,RHOS,US,PSIS,HS
685 READ(11,*) (EVIS(J),J = 1,IMAX)
ISTAG = 1
PSIS = 0.0
GO TO 500

C          AT EACH X WHERE PHYSICAL SPACE CALCULATIONS ARE DESIRED SAVE
C          1./(\RHO*U) ON EACH STREAMLINE
690 IF (XPST(LPS2).EQ.XPST(1)) ICNT = IPSI
JJJ = IPSI - ICNT + 1
IF (VAR(1).LT.XPST(LPS2)) GO TO 720
IF (VAR(1).GT.XPST(LPS2)) GO TO 705

C          VAR(1) = XPST(LPS2)
RHOURT = 1.0/(\RHO*U)
TEMP = TVI(1)
TEMPO = T
RCON1 = VAR(3)
RCON2 = VAR(4)
RCON3 = VAR(5)
RCON4 = VAR(6)
RCON5 = VAR(7)
RCON6 = VAR(8)
RCON7 = VAR(9)
IF (IMAX.EQ.10) THEN
RCON8 = VAR(10)
RCON9 = VAR(11)
RCON10 = VAR(12)
ENDIF
RRHO = RHO
WRITE(14,*) IPSI,RHOURT,VAR(1),PSIS
WRITE(16,*) LPS2,JJJ
IF (IMAX.EQ.7) WRITE(16,*) RCON1,RCON2,RCON3,RCON4,RCON5,
* RCON6,RCON7,RRHO,TEMP,TEMPO
IF (IMAX.EQ.10) WRITE(16,*) RCON1,RCON2,RCON3,RCON4,RCON5,RCON6
* RCON7,RCON8,RCON9,RCON10,RRHO,TEMP,TEMPO
GO TO 715
705 YPSIS = PREPSI + (XPST(LPS2) - PREX)*((PSIS - PREPSI)/
* (VAR(1) - PREX))
RHOURT = PRERU + (XPST(LPS2) - PREX)*((RHO*U - PRERU)/
* (VAR(1) - PREX))
TEMP = PRETEP + (XPST(LPS2) - PREX)*((TVI(1) - PRETEP)/
* (VAR(1) - PREX))
TEMPO = PRET + (XPST(LPS2) - PRET)*((T - PRET)/
* (VAR(1) - PRET))
RCON1 = PREC1 + (XPST(LPS2) - PREX)*((VAR(3) - PREC1)/
* (VAR(1) - PREX))
RCON2 = PREC2 + (XPST(LPS2) - PREX)*((VAR(4) - PREC2)/
* (VAR(1) - PREX))
RCON3 = PREC3 + (XPST(LPS2) - PREX)*((VAR(5) - PREC3)/
* (VAR(1) - PREX))
RCON4 = PREC4 + (XPST(LPS2) - PREX)*((VAR(6) - PREC4)/
* (VAR(1) - PREX))
RCON5 = PREC5 + (XPST(LPS2) - PREX)*((VAR(7) - PREC5)/
* (VAR(1) - PREX))
RCON6 = PREC6 + (XPST(LPS2) - PREX)*((VAR(8) - PREC6)/
* (VAR(1) - PREX))
RCON7 = PREC7 + (XPST(LPS2) - PREX)*((VAR(9) - PREC7)/
* (VAR(1) - PREX))
IF (IMAX.EQ.10) THEN

```

```

      RCON8 = PREC8 + (XPST(LPS2) - PREX)*((VAR(10) - PREC8)/
      *      (VAR(1) - PREX))
      RCON9 = PREC9 + (XPST(LPS2) - PREX)*((VAR(11) - PREC9)/
      *      (VAR(1) - PREX))
      RCON10 = PREC10 + (XPST(LPS2) - PREX)*((VAR(12) - PREC10)/
      *      (VAR(1) - PREX))
      ENDIF
      RRHO = PRERHO + (XPST(LPS2) - PREX)*((RHO - PRERHO)/
      *      (VAR(1) - PREX))
      RHOURT = 1.0/RHOURT
      WRITE(14,*) IPSI,RHOURT,XPST(LPS2),YPSIS
      WRITE(16,*) LPS2,JJJ
      IF (IMAX.EQ.7) WRITE(16,*) RCON1,RCON2,RCON3,RCON4,RCON5,
      *      RCON6,RCON7,RRHO,TEMP,TEMPO
      IF (IMAX.EQ.10) WRITE(16,*) RCON1,RCON2,RCON3,RCON4,RCON5,RCON6
      *      ,RCON7,RCON8,RCON9,RCON10,RRHO,TEMP,TEMPO
715 LPS2 = LPS2 + 1
      KSTAG = KSTAG + 1
720 PREX = VAR(1)
      PRERU = RHO*U
      PREPSI = PSIS
      PRETEP = TVI(1)
      PRET = T
      PREC1 = VAR(3)
      PREC2 = VAR(4)
      PREC3 = VAR(5)
      PREC4 = VAR(6)
      PREC5 = VAR(7)
      PREC6 = VAR(8)
      PREC7 = VAR(9)
      IF (IMAX.EQ.10) THEN
      PREC8 = VAR(10)
      PREC9 = VAR(11)
      PREC10 = VAR(12)
      ENDIF
      PRERHO = RHO
      IF (VAR(1).LT.VARI(IZTERM)) GO TO 505
C      END OF STREAMLINE
      WRITE(*,725)
      WRITE(8,725)
725 FORMAT(1X,'XXXXXXXXXXXXXXXXXX'/)
      IF ((IPSI.EQ.(NSR+1)).AND.(ISTAG.EQ.1)) GO TO 425
730 GO TO 420
C      COMPUTE PHYSICAL SPACE VALUES
735 WRITE(15,*) KSTAG
      CLOSE(10)
      CLOSE(11)
      CLOSE(12)
      CLOSE(13)
      CLOSE(15)
      CLOSE(16)
      WRITE(*,740) KSTAG
      WRITE(8,740) KSTAG
740 FORMAT(1X,'BEGIN PHYSICAL SPACE CALCULATIONS      KSTAG = ',I5/)
      OPEN(UNIT=17,FILE='RADIN3.DAT',STATUS='UNKNOWN')
C
      LPS3 = 0
      DO 840 LPS = 1,NXPST
      CLOSE(14)
      OPEN (UNIT=14,FILE='RHO RT.DAT',STATUS='OLD')

```

```

      IK = 0
745 LPS3 = LPS3 + 1
      IF (LPS3.GT.IZTERM) GO TO 840
      IF (DABS(X1(LPS3) - XPST(LPS)).GT.1.0E-08) GO TO 745
      DO 780 IPS = 1,KSTAG
      READ(14,*) IPSI,RHOURT,VAR(1),PSIS
      VARMX = VAR(1) - XPST(LPS)
      IF (DABS(VARMX).GT.1.0E-08) GO TO 780
      IK = IK + 1
      IBOB = IK - 1
      IF (IK.GT.200) WRITE(*,770) IK
770 FORMAT(1X,'IK.GT.200,CHANGE DIMENSION OF RUT AND PSISTG,IK = ',I3)
      RUT(IK) = RHOURT
      PSISTG(IK) = PSIS
780 CONTINUE
C                                     FIND SMALLEST DELTA PSI
      SMALL = 200.0
      DO 785 I = 2,IK
      IF ((PSISTG(I) - PSISTG(I-1)).LT.SMALL) SMALL = PSISTG(I)
      *   - PSISTG(I-1)
785 CONTINUE
      XDEL = PSISTG(IK)/SMALL
C                                     TRUNCATE
      L = INT(XDEL + 1.0)
C                                     REMAINDERING   M = 0 FOR EVEN, M = 1 FOR ODD
      M = MOD(L,2)
C                                     MAKE L EVEN
      IF (M.NE.0) L = L + 1
C                                     FIND INTEGRAL 1./(RHO*U) DELTA PSI FROM BODY TO SHOCK
C                                     USING SIMPSONS RULE ; L INCREMENTS , L+1 POINTS
      FL = L
      DPSIS = PSISTG(IK)/FL
      PSII = 0.0
      RB = RUT(1) + RUT(IK)
      DO 800 I = 2,L,2
      PSII = PSII + DPSIS
      CALL FTLUP (PSII,RU1,1,IK,PSISTG,RUT)
      IF (I.EQ.L) RB = RB + 4.0*RU1
      IF (I.EQ.L) GO TO 800
      PSII = PSII + DPSIS
      CALL FTLUP (PSII,RU2,1,IK,PSISTG,RUT)
      RB = RB + 4.0*RU1 + 2.0*RU2
800 CONTINUE
      RB = RB*DPSIS/3.0
      ARR = DSQRT(RSM(LPS3)**2 - 2.0*COSTM(LPS3)*RB)
      YI = (RSM(LPS3) - ARR)/COSTM(LPS3)
      WRITE(17,*) IBOB,IK
      WRITE(17,*) YI
      ZE = ZSM(LPS3) + YI*SINTM(LPS3)
      WRITE(*,815) PSISTG(1),X1(LPS3),ARR,YI,ZE
      WRITE(8,815) PSISTG(1),X1(LPS3),ARR,YI,ZE
815 FORMAT(1X,'PSI = ',E12.5,/,1X,'X = ',E13.6,3X,'R = ',E13.6
      *   ,3X,'Y = ',E13.6,3X,'Z = ',E13.6/)
      DO 830 I = 2,IK
      DPSI = PSISTG(I) - PSISTG(I-1)
C                                     TRAPEZOIDAL RULE
      TR = (DPSI/2.0)*(RUT(I) + RUT(I-1))
      RB = RB - TR
      ARR = DSQRT(RSM(LPS3)**2 - 2.0*COSTM(LPS3)*RB)
      YI = (RSM(LPS3) - ARR)/COSTM(LPS3)

```



```

WRITE(17,*) YI
ZE = ZSM(LPS3) + YI*SINTM(LPS3)
WRITE(8,815) PSISTG(I),X1(LPS3),ARR,YI,ZE
WRITE(*,815) PSISTG(I),X1(LPS3),ARR,YI,ZE
830 CONTINUE
WRITE(8,835)
WRITE(*,835)
835 FORMAT(//)
840 CONTINUE
845 WRITE(8,850)
WRITE(*,850)
850 FORMAT(1X,'END THIS CASE , HALLELUJAH '/')
CLOSE(8)
CLOSE(14)
CLOSE(17)
STOP
END

```

C

C

C

SUBROUTINE BASIC

C

C

C

C

```

BASIC CALLED BY CALINTH TO EVALUATE DERIVATIVES H DOT,
C SUB I DOT AND EV SUB I DOT ; DERIVATIVES START IN DER(1)

```

```

IMPLICIT REAL*8(A-H,O-Z)
COMMON /A1/ M,IX,IMAX,IPSI,MODEL,TE1,TE2
COMMON /A2/ P(200),DPDX(200),VARI(200)
COMMON /A3/ EVIINF(25),THETA1(25),MUI(25),FI(25),DELHI(25),
* CIINF(25),LI(25)
COMMON /A4/ R,MUINF,DELTA,LAMSQ,OMEGSQ,OMEGA,MU,T,U,TINF
COMMON /A5/ EPSIIL(20,25),GIL(20,25),CID(200,4)
COMMON /A6/ VAR(52),CUVAR(52),DER(51)
COMMON /A7/ MJ(50),EJ(50),AJ(50),BJ(50),DIRECT(50)
COMMON /A8/ TVI(25),NI(25),DGENI(25),BI(25)
COMMON /A10/ NUIJ(26,50),AIJ(25,50),NUIPJ(25,50)
COMMON /A11/ SIGIK(25,25),ALPIK(25,25),BETAIK(25,25)
COMMON /A13/ DELX,ZSTERM,IZTERM,NSR,MW,SP,TS
COMMON /A14/ EINF,PINF,RHOINF,VINF,E,JMAX,KEYINT,RHO,HSTAG
COMMON /A15/ PFTL,KITR1,NIP(25),UP(25)
COMMON /A16/ IUNEG,WE(25),WEXE(25),WEYE(25),WEZE(25)
COMMON /A20/ WE1,WE2
COMMON /A18/ ITNEG,IEXP
DIMENSION PHI(50),SJ(50),DCIJD(50),EVP(100)
DIMENSION CI(25),EVI(25),DCID(25),DEVID(25),EVIBAR(25)
EQUIVALENCE (CUVAR(2),H),(CUVAR(3),CI(1))
EQUIVALENCE (DER(1),DHDX),(DER(2),DCID(1))
REAL*8 KPJ,KEQ,KJ(50),LAMSQ,MU,MUI,MUIOT,MUINF,ND

```

C

C

C

C

C

C

C

C

C

C

C

C

```

DER(1) = DHDX          MUST BE DHDX AS H MAY BE + OR -
DER(2) = DCID(1)
-
-
DER(1+IMAX) = DCID(IMAX)
DER(1+IMAX+1) = DEVID(1)
DER(1+IMAX+IMAX) = DEVID(IMAX)
-
-
CUVAR(1) = X
CUVAR(2) = H,ENTHALPY - IT MUST BE H WHICH MAY BE + OR -
CUVAR(3) = CI(1)
-
-

```

```

C      CUVAR(2+IMAX)      - CI(IMAX)
C      CUVAR(2+IMAX+1)    - EVI(1)
C      CUVAR(2+IMAX+IMAX) - EVI(IMAX)
C
C                                     KEYINT - 1 AT END OF 1ST INTERVAL
      IF (ITNEG.EQ.1.OR.IEXP.EQ.1) RETURN
      IF (M.NE.0) GO TO 10
      DO 5 I = 1,IMAX
      J1 = I + IMAX + 2
5     CUVAR(J1) = VAR(J1)
C
C                                     INTERPOLATE FOR P ACROSS INTEGRATION INTERVAL
10    DO 15 I = 1,IMAX
15    EVI(I) = CUVAR(2+IMAX+I)
      MFTL = 1
      NFTL = IZTERM - IPSI + 1
      CALL FTLUP (CUVAR(1),PFTL,MFTL,NFTL,VARI(IPSI),P(IPSI))
C
C                                     INTERPOLATE FOR DPDX
      CALL FTLUP (CUVAR(1),DPFTL,MFTL,NFTL,VARI(IPSI),DPDX(IPSI))
      IF (KEYINT.EQ.0) GO TO 95
      SUM = 0.0
      DO 45 I = 1,IMAX
C
C                                     USE SHOCK VALUES FOR INITIAL COMPUTATION ON EACH STREAMLINE
      SUM = SUM + CI(I)/MUI(I)
      IF (EVI(I).EQ.0.0) TVI(I) = 0.0
      IF (EVI(I).EQ.0.0) GO TO 45
      XL = ((DGENI(I)*R*THETAI(I))/(MUI(I)*EVI(I)) + 1.0)
      ALN = DLOG(XL)
C
C                                     COMPUTE TVI
      TVI(I) = THETAI(I)/ALN
45    CONTINUE
      MU = 1.0/SUM
      U2 = 2.0*(HSTAG - H)
      IF (U2.LT.0.0) WRITE(*,50) H
50    FORMAT(1X,'U**2 IS NEG IN BASIC , H = ',E11.4,2X,
      * 'END STREAMLINE INTEGRATION')
      IF (U2.LT.0.0) IUNEG = 1
      IF (U2.LT.0.0) RETURN
      U = DSQRT(U2)
C
C                                     ITERATE FOR E
      KCODE = 0
55    KITR1 = 0
      ICODE = 0
      MAXI = 50
      TOL1 = .001
      TOL2 = .00001
C
      CALL FOFE (MAXI,TOL1,TOL2,ICODE,E)
C
      IF (ICODE.EQ.0) GO TO 75
      IF (ICODE.EQ.1) WRITE(*,60)
60    FORMAT(1X,'* MAXIMUM ITERATION EXCEEDED IN SUB BASIC *: STOP 66'//)
      IF (ICODE.EQ.2) WRITE(*,65)
65    FORMAT(1X,'*** DERIVATIVE = 0 IN SUB BASIC **** : STOP 66'//)
      WRITE(*,70) ICODE
70    FORMAT(1X,'ICODE = ',I2,' IN SUB BASIC '//)
      IF (ICODE.NE.1) STOP
      IF (KCODE.EQ.3) STOP
C
      WHEN ICODE = 1, TRY A NEW STARTING E. DO THIS 2 TIMES ;      STOP 66
      KCODE = KCODE + 1
      GO TO 55

```

```

75 IF (IEXP.NE.1) GO TO 85
   KCODE = KCODE + 1
   IEXP = 0
   IF (KCODE.EQ.1) GO TO 55
   IF (KCODE.GT.2) RETURN
   E = H - H*1.E-05
   GO TO 55
85 RHO = PFTL/(H - E)
   T = (PFTL*MU)/(RHO*R)
   IF (T.GT.0.0) GO TO 95
   WRITE(*,90) T,RHO,PFTL,H,E,MU
90 FORMAT(1X,'T NEGATIVE = ',E12.4,1X,'RHO = ',E11.4,1X,'PFTL = ',
*   E11.4,/,1X,'H = ',E11.4,1X,'E = ',E11.4,1X,'MU = ',E11.4,2X,
*   'IN SUB BASIC'/)
   E = H - H*1.0E-06
   ITNEG = 1
   RETURN
95 KEYINT = 1
   PI = 4.0*ATAN(1.0)
   BC = 1.38054E-16
   AVGN = 6.02252E+23
575 IF (IX.EQ.1) THEN
   TE = TINF
   TE1 = TE
   TE2 = TE
ENDIF

```

ELECTRON TEMPERATURE MODEL
ELECTRON TEMPERATURE MODEL

```

C
C
C   IF (IX.EQ.1) GO TO 576
C   CORR = 4.23E-06*(T**(-2.88))
C   CN = RHO*6.02252E+23
C   CNN = CN*VAR(5)/MUI(3)
C   CNO = CN*VAR(6)/MUI(4)
C   TE = DLOG(1. + 10.1/((CNO+CNN)*CORR))
C   TE = T/(1. + TE*T/85000.)
C   ITRTE=0
C   FNA=CN*CI(3) /MUI(3)
C   FNI=CN*CI(9) /MUI(9)
C   FNE=CN*CI(7) /MUI(7)
C   IF (FNE.LT.1.0D04)GO TO 576
C   FNM=CN*CI(1) /MUI(1)
C   EI=2.3322D-11
C   FAV=6.023D23
C   SA=1.52D-15
C   F1=0
C   F2=0
C   F0=0
C 574 ITRTE=ITRTE+1
C   IF(ITRTE.GT.1)GO TO 571
C   KK=0
C 569 KK=KK+1
C   IF(KK.GT.100)GO TO 576
C   TE=1000*KK
C 571 SM=0.5355D-19*TE+.696D-15
C   IF(ITRTE.GT.100)GO TO 573
C   SI=1.53614D08*(TE**3)/FNE
C   SI=DSQRT(SI)
C   SI=(4.38384D-06/(TE**2))*DLOG(SI)
C   FKF=1.1D32*(TE**(-3.14D0))*DEXP(-1.69D05/TE)
C   FKB=2.2D20*(TE**(-4.5D00))*1.D20

```

```

C      WEA=FKF*FNA-FKB*FNE*(FNI/FAV)
C      WEA=WEA*U*FAV/CI(7)
C      SX=FNA*SA+FNI*SI+FNMI/2.0D0*SM
C      W1=DCIDX(7)*U*FAV/CI(7)
C      WRITE(*,577)KK,ITRTE,W1,WEA,FE
C 577 FORMAT(2I5,2E11.3,F10.0)
C      W1=WEA
C      FE=T-1.23357D-10/SX/DSQRT(TE)*(WEA*EI+W1*3.45D-16*TE)-TE
C      IF(ITRTE.GT.1)GO TO 568
C      IF((FE/DABS(FE)*F0).LT.0.D00)GO TO 567
C      F0=FE
C      T0=TE
C      GO TO 569
C 567 F1=FE
C      T1=TE
C 566 TE=(T0*F1-T1*F0)/(F1-F0)
C      IF (DABS(TE-T1).LT.1.D00)GO TO 573
C      IF(DABS(TE-T0).LT.1.D00)GO TO 573
C      GO TO 574
C 568 IF((FE*F0).LT.0.0D00)GO TO 564
C      T0=TE
C      F0=FE
C      GO TO 566
C 564 F1=FE
C      T1=TE
C      GO TO 566
C573 CONTINUE
C 573 IF (TE.GT.T)TE=T
C      TE1 = TE
C      IF (ITRTE.GT.100)WRITE(*,572)
C 572 FORMAT(' ','TE DID NOT CONVERGE')
C      CN2 = RHO*6.02252E+23*VAR(3)/MUI(1)
C      A = DLOG(1.01D+01) - DLOG(CN2) - DLOG(6.4D+00)
C      *      - 59.*DLOG(1.0D+01) + 19.*DLOG(TVI(1))
C      A = DEXP(A)
C      TE = DLOG(1. + A)
C      TE = TVI(1)/(1. + TE*TVI(1)/85000.)
C      TE2 = TE
C      576 CONTINUE
C
C      COMPUTE DCIDX AND DEVIDX FOR EACH SPECIE I
C
C      DO 210 I = 1,IMAX
C      DCIDX(I) = 0.0
C      CVSUM1 = 0.0
C      CVSUM2 = 0.0
C
C      CORRECTED RELAXATION TIME FOR PARK MODEL
C      IF (MODEL.EQ.4.AND.I.EQ.1) THEN
C      ND = RHO*AVGN*CI(I)/MUI(I)
C      SIGMAV = 1.00E-17*(50000./T)**2
C      C = DSQRT(8.*AVGN*BC*T/(PI*MUI(I)))
C      TAUC = 1.0/(C*SIGMAV*ND)
C      ENDIF
C
C      FOR EACH REACTION J
C
C      DO 190 J = 1,JMAX
C      ICT = MOD(J,2)
C
C      II IS THE SELECTED SPECIE FOR REACTION J
C      KNT = 0
C      IF (ICT.EQ.1) THEN
C      IF (DIRECT(J).EQ.1.0) PHIC = 1.0
C      IF (DIRECT(J).EQ.2.0) PHIC = 2.0

```

```

      II = MJ(J)
100  IF (M.EQ.0) GO TO 115
      IF (FI(II).NE.0.0) GO TO 120
115  IF (PHIC.EQ.1.0) PHI(J) = 1.0
      IF (PHIC.EQ.2.0) PHI(J+1) = 1.0
      GO TO 125
120  IF (MODEL.EQ.3.AND.UP(II).NE.0.0) GO TO 122
      TEM = DEXP(THETAI(II)/TVI(II))
      TEM1 = DEXP(THETAI(II)/T)
      TEM2 = DEXP(-NI(II)*(THETAI(II)/TVI(II) - THETAI(II)/T))
      TEM3 = DEXP(THETAI(II)/TVI(II) - THETAI(II)/T)
      IF (TEM3.EQ.1.0.AND.PHIC.EQ.1.0) PHI(J) = 1.0
      IF (TEM3.EQ.1.0.AND.PHIC.EQ.2.0) PHI(J+1) = 1.0
      IF (TEM3.EQ.1.0) GO TO 125
C      PHI = COUPLING COEFF.; FOR CVD,CVDV AND PARK MODEL
      IF (PHIC.EQ.1.0) PHI(J) = ((1.0-TEM2)/(TEM3-1.0)*(TEM - 1.0)
*      /(TEM1 - 1.0))/NI(II)
      IF (PHIC.EQ.2.0) PHI(J+1) = ((1.0-TEM2)/(TEM3-1.0)*(TEM - 1.0)
*      /(TEM1 - 1.0))/NI(II)
      GO TO 125
122  USUM = 0.0
      TSUM = 0.0
      TFSUM = 0.0
      TVTSUM = 0.0
      TF = 1.0/(1./TVI(II) - 1./T - 1./UP(II))
      NIC = NIP(II) + 1
      DO 123 N = 1,NIC
      XN = N
      EEVP = XN - 0.5
      EVP(N) = EEVP*(WE(II) + EEVP*(-WEZE(II) + EEVP*(WEYE(II) +
*      WEZE(II)*EEVP)))
      EVP(N) = EVP(N) - EVP(1)
      USUM = USUM + DEXP(EVP(N)/UP(II))
      TSUM = TSUM + DEXP(-EVP(N)/T)
      TFSUM = TFSUM + DEXP(-EVP(N)/TF)
123  TVTSUM = TVTSUM + DEXP(-EVP(N)/TVI(II))
C      COUPLING COEFF. FOR CVDV-PREFERENTIAL MODEL
      IF (PHIC.EQ.1.0) PHI(J) = (TSUM*TFSUM)/(USUM*TVTSUM)
      IF (PHIC.EQ.2.0) PHI(J+1) = (TSUM*TFSUM)/(USUM*TVTSUM)
125  KNT = KNT + 1
      IF (KNT.EQ.1.AND.PHIC.EQ.1.0) THEN
        PHIC = 2.0
        II = MJ(J+1)
        GO TO 100
      ENDIF
      IF (KNT.EQ.1.AND.PHIC.EQ.2.0) THEN
        PHIC = 1.0
        II = MJ(J+1)
        GO TO 100
      ENDIF
C      START OF LOOP TO DETERMINE EQUILIBRIUM CONSTANT
      BETAJ = 0.0
      FJORT = 0.0
      DO 130 K = 1,IMAX
      SUMG = 0.0
      LII = LI(K)
      DO 128 L = 1,LII
128  SUMG = SUMG + GIL(L,K)*DEXP(-EPSIIL(L,K)/T)
C      MUIOT = (CHEMICAL POTENTIAL)/T
      MUIOT = - (BI(K) + (5. + 2.*FI(K))/2.*(DLOG(TINF) +

```

```

*      DLOG(T/TINF)) + FI(K)*DLOG(1./(1. - DEXP(-THETAI(K)/T)))
*      + DLOG(SUMG/GIL(1,K))) + DELHI(K)/(R*T)
BETAJ = BETAJ + (NUPIJ(K,J) - NUIJ(K,J))
130 FJORT = FJORT + (NUPIJ(K,J) - NUIJ(K,J))*MUIOT
C      KPJ = EQUIL. CONST. (IN TERMS OF PARTIAL PRESSURES)
      KPJ = DEXP(-FJORT)
C      KEQ = EQUILIBRIUM CONSTANT
      KEQ = KPJ*(9.8688225E-07*R*T)**(-BETAJ)
C      KJ = RATE CONSTANT (FORW: J = 1,3,.. BACK: J = 2,4,..)
      IF (DIRECT(J).EQ.1.0) THEN
        KJ(J) = AJ(J)*T**BJ(J)*DEXP(-EJ(J)/T)
        IF (J.EQ.19) KJ(J) = AJ(J)*TE1 **BJ(J)*DEXP(-EJ(J)/TE1 )
        IF (J.EQ.21) KJ(J) = AJ(J)*TE1 **BJ(J)*DEXP(-EJ(J)/TE1 )
        IF (TE1.LT.2.D3)GO TO 131
        IF(J.EQ.19) KEQ=(1.8D-08)*(TE1**1.5D0)*DEXP(-1.69D5/TE1)
        IF(J.EQ.21) KEQ=(3.59D-09)*(TE1**1.5D0)*DEXP(-1.575D5/TE1)
131  KJ(J+1) = KJ(J)/KEQ
      ELSE
        KJ(J+1) = AJ(J)*T**BJ(J)*DEXP(-EJ(J)/T)
        IF (J.EQ.19) KJ(J+1) = AJ(J)*TE1 **BJ(J)*DEXP(-EJ(J)/TE1 )
        IF (J.EQ.21) KJ(J+1) = AJ(J)*TE1 **BJ(J)*DEXP(-EJ(J)/TE1 )
        IF(TE1.LT.2.D3)GO TO 132
        IF(J.EQ.19) KEQ=(1.8D-08)*(TE1**1.5D0)*DEXP(-1.69D5/TE1)
        IF(J.EQ.21) KEQ=(3.59D-09)*(TE1**1.5D0)*DEXP(-1.575D5/TE1)
132  KJ(J) = KJ(J+1)*KEQ
      ENDIF
      ENDIF
      IF (NUIJ(IMAX+1,J).EQ.0) SJ(J) = 1.0
      IF (NUIJ(IMAX+1,J).EQ.0) GO TO 140
      SUM = 0.0
      DO 135 ISUM = 1,IMAX
135  SUM = SUM + AIJ(ISUM,J)*CI(ISUM)/MUI(ISUM)
      SJ(J) = RHO*SUM
140  PROD = 1.0
      DO 180 IPROD = 1,IMAX
      IF (NUIJ(IPROD,J).EQ.0) GO TO 180
      TEM = RHO*CI(IPROD)/MUI(IPROD)
      IF (TEM.GE.0.0) GO TO 165
C  LET PROBLEM COMPUTE PROD. WHEN CI NEG ; DECIDE IN MAIN ABOUT ACCEPTING NEG CI
      LABW = MOD(NUIJ(IPROD,J),2)
C      LABW = 0 FOR EVEN NUIJ, ABW = 1 FOR ODD NUIJ
      IF (LABW.EQ.0) PROD = PROD*(-TEM)**NUIJ(IPROD,J)
      IF (LABW.EQ.1) PROD = -1.0*PROD*(-TEM)**NUIJ(IPROD,J)
      GO TO 180
165  PROD = PROD*TEM**NUIJ(IPROD,J)
180  CONTINUE
      IF (ICT.EQ.1) PRODF = PROD
      BETAIJ = NUPIJ(I,J) - NUIJ(I,J)
      DCIJDX(J) = PHI(J)*KJ(J)*SJ(J)*(MUI(I)/(RHO*U))*PROD*BETAIJ
C      DCIDX = RATE OF PRODUCTION OF THE CONCENTRATION OF SPECIE I
      DCIDX(I) = DCIDX(I) + DCIJDX(J)
      IF (ICT.EQ.0.AND.MODEL.GT.1) THEN
C      QIJ = NET RATE OF PRODUCTION
        QIJ = DCIJDX(J-1) + DCIJDX(J)
        DENOM = KEQ*PHI(J-1)*SJ(J-1)*PRODF
        IF (DENOM.EQ.0.0) GO TO 190
C      CHI = DEGREE OF NON-EQUILIBRIUM
        CHI = 1.0 - (PHI(J)*SJ(J)*PROD/DENOM)
        DENOM1 = CI(I)*CHI
        IF (DENOM1.EQ.0.0) GO TO 190

```

```

      CVSUM1 = CVSUM1 + QIJ/DENOM1
      CVSUM2 = CVSUM2 + QIJ*(1.0 - CHI)/DENOM1
    ENDIF
190 CONTINUE
    IF(I.EQ.7)WE1=DCIJDX(19)+DCIJDX(20)
    IF(I.EQ.7)WE2=DCIJDX(21)+DCIJDX(22)
C    IF (I.EQ.7)WRITE(*,191)WE1
191 FORMAT(' WE1 = ',E13.3)
C
    TEM = DEXP(THETAI(I)/T)
    TEM1 = (FI(I)*DGENI(I)*R*THETAI(I))/MUI(I)
    EVIBAR(I) = TEM1/(TEM - 1.0)
    IF (M.NE.0) GO TO 195
    VAR(I+IMAX+2) = EVIBAR(I)
    DEVIDX(I) = 0.0
    GO TO 210
195 TAUSUM = 0.0
    IF (FI(I).EQ.0.0) GO TO 210
C
CVD MODEL
    DO 200 K = 1,IMAX
    TEM3 = DEXP(-THETAI(I)/T)
    TEM4 = DEXP(SIGIK(K,I)*T**(-1./3.))
    TEM5 = (FI(I)*ALPIK(K,I))/PFTL
    TAUIC = TEM5*(T**BETAIK(K,I)*TEM4)/(1.0 - TEM3)
C
MILLIKAN & WHITE RELAXATION DATA
    IF (MW.EQ.1) THEN
      IF (I.EQ.1) TAUIC = 1.9E-05*DEXP(216.45*T**(-1./3.))/PFTL
      IF (I.EQ.8) TAUIC = 3.1845E-05*DEXP(198.66*T**(-1./3.))/PFTL
    ENDIF
    TEM6 = CI(K)*(EVIBAR(I) - EVI(I))
C
PARK MODEL
    IF (MODEL.EQ.4.AND.I.EQ.1) THEN
      TAULP = TAUIC + TAUC
      PF = DABS((T - TVI(I))/(TS - TINF))**(SP - 1.)
      TAUSUM = TAUSUM + (TEM6/(TAULP*U))*PF
      GO TO 200
    ENDIF
    TAUSUM = TAUSUM + TEM6/(TAUIC*U)
200 CONTINUE
    IF (MODEL.EQ.1) GO TO 207
    IF (MODEL.EQ.3) GO TO 202
C
ADDITIONAL TERMS FOR CVDV & PARK MODEL
    IF (I.EQ.5.OR.I.EQ.6.OR.I.EQ.8) GO TO 207
    TEM7 = DEXP(THETAI(I)/TVI(I) - THETAI(I)/T)
    TEM8 = DEXP(NI(I)*THETAI(I)/TVI(I))*DEXP(-NI(I)*THETAI(I)/T)
    CVDVT = ((THETAI(I)/(TEM7 - 1.) - NI(I)*THETAI(I)/(TEM8 - 1.))
    *      *R/MUI(I) - EVI(I))*CVSUM1 - ((0.5*(NI(I) - 1.)*THETAI(I)
    *      *R/MUI(I)) - EVI(I))*CVSUM2
    TAUSUM = TAUSUM + CVDVT
    GO TO 207
C
ADDITIONAL TERMS FOR CVDV-PREFERENTIAL MODEL
202 IF (I.EQ.5.OR.I.EQ.6.OR.I.EQ.8) GO TO 207
    USUM1 = 0.0
    USUM2 = 0.0
    TFSUM1 = 0.0
    TFSUM2 = 0.0
    NIC = NIP(I) + 1
    TF = 1.0/(1./TVI(I) - 1./T - 1./UP(I))
    DO 204 N = 1,NIC
    XN = N

```

```

      EEVP = XN - 0.5
      EVP(N) = EEVP*(WE(I) + EEVP*(-WEXE(I) + EEVP*(WEYE(I) +
*          WEZE(I)*EEVP)))
      EVP(N) = EVP(N) - EVP(1)
      USUM1 = USUM1 + DEXP(EVP(N)/UP(I))
      USUM2 = USUM2 + EVP(N)*DEXP(EVP(N)/UP(I))
      TFSUM1 = TFSUM1 + DEXP(-EVP(N)/TF)
204 TFSUM2 = TFSUM2 + EVP(N)*DEXP(-EVP(N)/TF)
      GBAR = (USUM2/USUM1)*(R/MUI(I))
      EBAR = (TFSUM2/TFSUM1)*(R/MUI(I))
      CVDVPT = (EBAR - EVI(I))*CVSUM1 - (GBAR - EVI(I))*CVSUM2
      TAUSUM = TAUSUM + CVDVPT
C                                     DEVIDX = EQUILIBRIUM VIBRATIONAL ENERGY
207 DEVIDX(I) = TAUSUM
      DER(1+IMAX+I) = DEVIDX(I)
210 CONTINUE
C                                     COMPUTE DHDX
      DHDX = DPFTL/RHO
      RETURN
      END
C
C-----
C
      SUBROUTINE FOFE (MAXI,TOL1,TOL2,ICODE,E1)
C
C CALLED BY SUB BASIC TO EVALUATE E BY NEWTON ITERATION TECHNIQUE
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /A1/ M,IX,IMAX,IPSI,MODEL,TE1,TE2
      COMMON /A2/ P(200),DPDX(200),VARI(200)
      COMMON /A3/ EVIINF(25),THETA1(25),MUI(25),FI(25),DELHI(25),
*          CIINF(25),LI(25)
      COMMON /A4/ R,MUINF,DELTA,LAMSQ,OMEGSQ,OMEGA,MU,T,U,TINF
      COMMON /A5/ EPSIIL(20,25),GIL(20,25),CID(200,4)
      COMMON /A6/ VAR(52),CUVAR(52),DER(51)
      COMMON /A7/ MJ(50),EJ(50),AJ(50),BJ(50),DIRECT(50)
      COMMON /A8/ TVI(25),NI(25),DGENI(25),BI(25)
      COMMON /A10/ NUIJ(26,50),AIJ(25,50),NUPIJ(25,50)
      COMMON /A15/ PFTL,KITR1,NIP(25),UP(25)
      COMMON /A18/ ITNEG,IEXP
      DIMENSION CI(25)
      EQUIVALENCE (CUVAR(3),CI(1)),(CUVAR(2),H)
      REAL*8 MUI,LAMSQ,MUINF,MU
C
      IF (IEXP.EQ.1.OR.ITNEG.EQ.1) RETURN
      KITR1 = KITR1 + 1
      ITER = 0
      PORHO = H - E1
      T = PORHO*MU/R
1  ITER = ITER + 1
      E = 0.0
      DE = 0.0
      DO 35 I = 1,IMAX
      IF (TVI(I).EQ.0.0) THEN
          TEM1 = (1.5*R*T)/MUI(I) + (FI(I)*R*T)/MUI(I)
          DTEM1 = 1.5*R/MUI(I) + FI(I)*R/MUI(I)
          GO TO 15
      ENDIF
      TEM = DEXP(THETA1(I)/TVI(I))
      TEM1 = (1.5*R*T)/MUI(I) + (FI(I)*R*T)/MUI(I) +

```



```

      *      (FI(I)*R*THETAI(I))/(MUI(I)*(TEM - 1.0))
      DTEM1 = 1.5*R/MUI(I) + FI(I)*R/MUI(I) + (R*FI(I)*THETAI(I)**2)
      *      /(MUI(I)*TVI(I)**2)*(TEM/(TEM - 1.0)**2)
15  SUMG = 0.0
      SUMGE = 0.0
      DSUMGE = 0.0
      LII = LI(I)
C
      DO 25 L = 1,LII
      TEM3 = - EPSIIL(L,I)/T
      IF (TEM3.LT.741.67) GO TO 22
      IEXP = 1
      RETURN
22  TEM2 = DEXP(TEM3)
      SUMG = SUMG + GIL(L,I)*TEM2
      SUMGE = SUMGE + GIL(L,I)*EPSIIL(L,I)*TEM2
      IF (SUMGE.LT.1.0D+34) GO TO 25
      IEXP = 1
      RETURN
25  DSUMGE = DSUMGE + TEM2*GIL(L,I)*EPSIIL(L,I)**2
      DEVI = R/MUI(I)*(((SUMG*DSUMGE/T**2) - (SUMGE/T)**2)
      *      /SUMG**2)
      DEI = DTEM1 + DEVI
      EI = TEM1 + R/MUI(I)*(SUMGE/SUMG) + DELHI(I)/MUI(I)
      DE = DE + DEI*CI(I)
      E = E + EI*CI(I)
35  CONTINUE
C
      IF (E.GT.H) ITNEG = 1
      DFT = - DE
      FT = E1 - E
      IF (DFT.EQ.0.0) THEN
      ICODE = 2
      RETURN
      ENDIF
      TS1 = T - (FT/DFT)
      DELT = DABS(TS1 - T)
      T = TS1
      E1 = H - R*T/MU
      IF (ITER.GE.MAXI) THEN
      ICODE = 1
      RETURN
      ENDIF
      IF (DELT.GT.TOL1) THEN
      ITER1 = ITER
      GO TO 1
      ENDIF
      ITER2 = ITER1 + 10
      IF (DELT.LE.TOL2.OR.ITER.GE.ITER2) RETURN
      GO TO 1
      END
C
C -----
C
      SUBROUTINE FOFTS1 (MAXI,TOL1,TOL2,ICODE,TSG)
C
C CALLED BY MAIN TO EVALUATE TS BY NEWTON ITERATION TECHNIQUE
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /A1/ M,IX,IMAX,IPSI,MODEL,TE1,TE2

```

```

COMMON /A2/ P(200),DPDX(200),VARI(200)
COMMON /A3/ EVIINF(25),THETAI(25),MUI(25),FI(25),DELHI(25)
*      ,CIINF(25),LI(25)
COMMON /A4/ R,MUINF,DELTA,LAMSQ,OMEGSQ,OMEGA,MU,T,U,TINF
COMMON /A5/ EPSIIL(20,25),GIL(20,25),CID(200,4)
COMMON /A6/ VAR(52),CUVAR(52),DER(51)
REAL*8 MUI,LAMSQ,MUINF,MU

```

C

```

ITER = 0
1 ITER = ITER + 1
ES = 0.0
DES = 0.0
DO 25 I = 1,IMAX
IF (M.EQ.1) EVIS = EVIINF(I)
IF (M.EQ.1) GO TO 5
TEM = DEXP(THETAI(I)/TSG)
EVIS = (R*THETAI(I))/(MUI(I)*(TEM-1.0))*FI(I)
DEVIS = (R*FI(I)*THETAI(I)**2)/(MUI(I)*TSG**2)*(TEM/(TEM-1.0)**2)
5 SUMG = 0.0
SUMGE = 0.0
DSUMGE = 0.0
LII = LI(I)

```

C

```

DO 10 L = 1,LII
TEM1 = DEXP(- EPSIIL(L,I)/TSG)
SUMG = SUMG + TEM1*GIL(L,I)
SUMGE = SUMGE + TEM1*GIL(L,I)*EPSIIL(L,I)
10 DSUMGE = DSUMGE + TEM1*GIL(L,I)*EPSIIL(L,I)**2
EEIS = (R/MUI(I))*(SUMGE/SUMG)
DEEIS = R/MUI(I)*(((SUMG*DSUMGE/TSG**2) - (SUMGE/TSG)**2)
*      /SUMG**2)
EIS = (1.5*R*TSG)/MUI(I) + (FI(I)*R*TSG)/MUI(I) + EVIS + EEIS
*      + DELHI(I)/MUI(I)
DEIS = 1.5*R/MUI(I) + FI(I)*R/MUI(I) + DEVIS + DEEIS
ES = ES + EIS*CIINF(I)
DES = DES + DEIS*CIINF(I)
25 CONTINUE

```

C

```

FAC = DSQRT(OMEGSQ - 2.0*LAMSQ*(DELTA - ES))
FIN = (2.0*MUINF*(DELTA - ES)*FAC)/(R*(OMEGA + FAC))
FTS = TSG - FIN
FAC1 = R*(OMEGA+FAC)*(MUINF*(DELTA-ES)*(2.0/FAC)*LAMSQ*DES -
*      2.0*FAC*MUINF*DES) - 2.0*MUINF*(DELTA-ES)*R*LAMSQ*DES
FAC2 = R**2*(OMEGA + FAC)**2
DFTS = 1.0 - FAC1/FAC2
IF (DFTS.EQ.0.0) THEN
ICODE = 2
RETURN
ENDIF
TSG1 = TSG - (FTS/DFTS)
DELTS = DABS(TSG1 - TSG)
TSG = TSG1
IF (ITER.GE.MAXI) THEN
ICODE = 1
RETURN
ENDIF
IF (DELTS.GT.TOL1) THEN
ITER1 = ITER
GO TO 1
ENDIF

```



```

      IEXP = 0
      RETURN
25  IF (DABS((HPREV - H)/H).GT.HCHECK) WRITE(*,27)
27  FORMAT(1X,'CHECK ABS((HPREV-H)/H).GT.HCHECKT IN SUB CHECK'/)
      IF (DABS((HPREV - H)/H).GT.HCHECK) GO TO 15
C                                     ACCEPTABLE
30  TPREV = T
      HPREV = H
      ELB = 0.0
      RETURN
      END

C
C-----
C
      SUBROUTINE SHOCKG (NXPSTM1,ITK)
C
C  SUBROUTINE SHOCKG CALLED BY MAIN AND MUST BE SUPPLIED BY USER.
C  X,ZS,RS,RC,COST,SINT FOR EACH X FROM 0.0 TO X AT ZTERM IN INCREMENTS
C  OF (DELX/NSR) TO DELX AND INCREMENTS OF DELX THEREAFTER.
C
C      X          -   DISTANCE ALONG SHOCK
C      ZS         -   DISTANCE ALONG SHOCK AXIS OF SYMMETRY
C      RS         -   RADIUS OF SHOCK
C      RC         -   RADIUS OF CURVATURE OF SHOCK
C      COST       -   COS OF ANGLE OF ATTACK
C      SINT       -   SIN OF ANGLE OF ATTACK
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /A9/  EVIS(25),XPST(100)
      COMMON /A12/ SINTM(200),COSTM(200),RSM(200),RCM(200),X1(200)
      *              ,ZSM(200)
      COMMON /A13/ DELX,ZSTERM,IZTERM,NSR,MW,SP,TS
      DIMENSION ZDUM(501),RSDUM(501),RCDUM(501),XC(501),CODUM(501),
      *          SIDUM(501)
C      Rs/L = SQRT(2*R*Zs/L - Bs*(Zs/L)**2)      WHERE: L = 1.0 cm
      ITK = 0
      EL = 1.0
      R = 230.0
      BS = -4.0
      DO 5 I = 2,501
5  ZDUM(I) = (I-1)*ZSTERM/500
      ZDUM(1) = 0.0
      RSDUM(1) = 0.0
      XC(1) = 0.0
      CODUM(1) = 0.0
      SIDUM(1) = 1.0
      RCDUM(1) = 230.0
C
      DO 10 I = 2,501
      ZSND = ZDUM(I)/EL
      RSDUM(I) = DSQRT(2.*R*ZSND - BS*ZSND**2)
      C = 1.0/(R - BS*ZSND)**2
      C1 = DSQRT(C*RSDUM(I)**2 + 1.0)
      XC(I) = RSDUM(I)/2.*C1 + 1./(2.*DSQRT(C))*DLOG(RSDUM(I)*DSQRT(C)
      *      + C1)
      D1 = (R - BS*ZSND)/RSDUM(I)
      D2 = - (R - BS*ZSND)**2/RSDUM(I)**3 - BS/RSDUM(I)
      THETA = ATAN(D1)
      SIDUM(I) = DSIN(THETA)
      CODUM(I) = DCOS(THETA)

```

```

      RCDUM(I) = (1.0 + D1**2)**1.5/DABS(D2)
C
      RCDUM(I) = EL*RCDUM(I)
      XC(I) = EL*XC(I)
      RSDUM(I) = EL*RSDUM(I)
10  CONTINUE
C
      WRITE(*,25)
      WRITE(8,25)
25  FORMAT(5X,'X',11X,'ZS',10X,'RS',10X,'RC',9X,'COST',8X,'SINT'/)
      NOX = 0
      M = 1
      NZS = 501
C
      INTERPOLATE SHOCK VALUES TO X SHOCK COORDINATE
30  NOX = NOX + 1
      IF (NOX.LT.(NSR+2)) X = X + DELX/NSR
      IF (NOX.EQ.1) X = 0.0
      IF (NOX.GT.(NSR+1)) X = X + DELX
      CALL FTLUP (X,ZS,M,NZS,XC,ZDUM)
      CALL FTLUP (X,RS,M,NZS,XC,RSDUM)
      CALL FTLUP (X,RC,M,NZS,XC,RCDUM)
      CALL FTLUP (X,COST,M,NZS,XC,CODUM)
      CALL FTLUP (X,SINT,M,NZS,XC,SIDUM)
      WRITE(8,35) X,ZS,RS,RC,COST,SINT,NOX
      WRITE(*,35) X,ZS,RS,RC,COST,SINT,NOX
35  FORMAT(1X,6(E10.4,2X),I4)
      SINTM(NOX) = SINT
      COSTM(NOX) = COST
      RSM(NOX) = RS
      RCM(NOX) = RC
      X1(NOX) = X
      ZSM(NOX) = ZS
      DO 36 IT = 2,NXPSTM1
      IF (DABS(XPST(IT) - X).GT.1.0E-06) GO TO 36
      ITK = ITK + 1
36  CONTINUE
      IF (ZS.GE.0.0) GO TO 40
      WRITE(*,37) ZS
37  FORMAT(1X,'ZS SHOULD BE GREATER THAN 0.0 : ZS = ',E10.5,
*      /,1X,'STOP 13'/)
      STOP
C
      STOP 13
40  IF (ZS.LT.ZSTERM) GO TO 30
      IZTERM = NOX
      RETURN
      END
C
C-----
C
      SUBROUTINE CALINTH (N,CIMAX,PHMAX)
C
C      IN THE CALINTH VERSION OF CALINT, THE VARIABLE IN VAR(2) AND
C      CUVAR(2) MAY BE + OR -. VALUES OF OTHER DEPENDENT VARIABLES
C      ARE EXPECTED TO BE POSITIVE.
C
C      10-69
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /A1/ M,IX,IMAX,IPSI,MODEL,TE1,TE2
      COMMON /A2/ P(200),DPDX(200),VARI(200)
      COMMON /A3/ EVIINF(25),THETAI(25),MUI(25),FI(25),DELHI(25),
*      CIINF(25),LI(25)

```

```

COMMON /A4/ R,MUINF,DELTA,LAMSQ,OMEGSQ,OMEGA,MU,T,U,TINF
COMMON /A5/ EPSIIL(20,25),GIL(20,25),CID(200,4)
COMMON /A6/ VAR(52),CUVAR(52),DER(51)
COMMON /A7/ MJ(50),EJ(50),AJ(50),BJ(50),DIRECT(50)
COMMON /A8/ TVI(25),NI(25),DGENI(25),BI(25)
COMMON /A10/ NUIJ(26,50),AIJ(25,50),NUIPJ(25,50)
COMMON /A11/ SIGIK(25,25),ALPIK(25,25),BETAIK(25,25)
COMMON /A17/ ELB,SPEC,CJ,TPREV,HPREV,HCHECK,TCHECK
COMMON /A19/ ELE1(51),ELE2(51),NERR
REAL*8 MUI,MU,MUINF,LAMSQ
DIMENSION F1(51),F2(51),F3(51),CAPF1(51),CAPF2(51),CAPF3(51)
*      ,P1(51),PH(51),DELTY(51),Y3(51),Y4(51),F4(51),Y2(51)

```

```

C
C IF CJ OR N IS EQUAL TO ZERO :          NERR = 1
C IF ELE1 IS LESS THAN OR EQUAL TO ELE2 : NERR = 2
C                                          TEST INPUT

```

```

      FN = N
      TEST = CJ*FN
      IF (TEST) 998,997,998
997  NERR = 1
      RETURN
998  DO 999 I = 1,N
      IF ((ELE1(I) - ELE2(I)).LE.0.0) THEN
        NERR = 2
        RETURN
      ENDIF
999  CONTINUE
1000 IF (SPEC) 5,1,5

```

```

C          SECTION FOR INITIALIZATION COMPUTATION OF DERIVATIVES

```

```

1  SPEC = CJ
   ICONT = 1
2  N1 = N + 1
   DO 3 I = 1,N1
3  CUVAR(I) = VAR(I)
   CALL BASIC

```

```

C          RETURN WITH DERIVATIVES IN DER

```

```

      DO 4 I = 1,N
4  F1(I) = DER(I)
   RETURN

```

```

C          COMPUTE Y2,X2

```

```

5  CUVAR(1) = VAR(1) + CJ/2.0
   DO 6 I = 1,N
   I1 = I + 1
   Y2(I) = VAR(I1) + CJ/2.0*F1(I)
   IF (I.EQ.1) GO TO 6
   IF (Y2(I)) 65,6,6
6  CUVAR(I1) = Y2(I)
   GO TO 66
65  SPEC = CJ
   CJ = CJ/2.0
   GO TO 5

```

```

C          CALL BASIC TO EVALUATE F2

```

```

66  CALL BASIC

```

```

C          RETURN

```

```

      DO 7 I = 1,N
      I1 = I + 1
      F2(I) = DER(I)

```

```

C          COMPUTE Y3

```

```

      Y3(I) = VAR(I1) + CJ/2.0*F2(I)
      IF (I.EQ.1) GO TO 7

```

```

      IF (Y3(I)) 65,7,7
7    CUVAR(I1) = Y3(I)
C
      CALL BASIC
      CALL BASIC
C
      RETURN
      DO 10 I = 1,N
      F3(I) = DER(I)
C
      COMPUTE P,PH AND CAP F TERMS
      IF (Y3(I) - Y2(I)) 9,8,9
8    P1(I) = 0.0
      GO TO 91
9    P1(I) = -((F3(I) - F2(I))/(Y3(I) - Y2(I)))
91   PH(I) = P1(I)*CJ
      IF (PH(I)) 83,83,103
83   PH(I) = 0.0
      P1(I) = 0.0
      GO TO 84
103  Z1 = DABS(Y3(I) - Y2(I))/((DABS(Y3(I)) + DABS(Y2(I)))/2.0)
      IF (Z1 - 0.5E-04) 83,83,84
84   IF (PH(I) - 0.1) 85,85,95
85   CAPF1(I) = 1.0 - PH(I)/2.0 + PH(I)**2/6.0 - PH(I)**3/24.0
      CAPF2(I) = 0.5 - PH(I)/6.0 + PH(I)**2/24.0 - PH(I)**3/120.0
      CAPF3(I) = 1./6. - PH(I)/24. + PH(I)**2/120. - PH(I)**3/720.
      GO TO 10
95   CAPF1(I) = (DEXP(-PH(I)) - 1.0)/(- PH(I))
      CAPF2(I) = (CAPF1(I) - 1.0)/(- PH(I))
      CAPF3(I) = (CAPF2(I) - 0.5)/(- PH(I))
10  CONTINUE
C
      IS PH BETWEEN ELE2 AND ELE1
      IF (ICONT - 1) 101,101,102
102  ICONT = ICONT - 1
      SPEC = CJ
      GO TO 17
101  DO 11 I = 1,N
      IF (DABS(PH(I)) - ELE1(I)) 11,11,13
11  CONTINUE
      SPEC = CJ
      GO TO 15
C
      HALVE INTERVAL AND DOUBLE PH RANGE
13  DO 96 I = 1,N
      ELE1(I) = ELE1(I)*2.0
      IF (ELE1(I) - PHMAX) 94,94,955
94  ELE2(I) = ELE2(I)*2.0
      GO TO 96
955  ELE1(I) = ELE1(I)/2.0
96  CONTINUE
      SPEC = CJ
      CJ = CJ/2.0
      ICONT = 3
      GO TO 5
C
      RETURN TO RECOMPUTE INTERVAL
15  DO 16 I = 1,N
      IF (DABS(PH(I)) - ELE2(I)) 16,17,17
16  CONTINUE
C
      DOUBLE INTERVAL
      CJ = 2.0*CJ
      IF (CJ - CIMAX) 17,17,165
165  CJ = CIMAX
C
      COMPUTE Y4,X4
17  DO 18 I = 1,N

```

```

      I1 = I + 1
      CUVAR(I1) = VAR(I1) + SPEC*(F3(I)*(2.0*CAPF2(I)) + F1(I)*
*      (CAPF1(I) - 2.0*CAPF2(I)) + F2(I)*PH(I)*CAPF2(I))
      IF (I.EQ.1) GO TO 18
      IF (CUVAR(I1)) 175,18,18
175  CJ = SPEC
      CJ = CJ/2.0
      GO TO 5
18   Y4(I) = CUVAR(I1)
      CUVAR(1) = VAR(1) + SPEC
C
C      CALL BASIC TO EVALUATE F4
      CALL BASIC
C
C      RETURN
      DO 20 I = 1,N
      I1 = I + 1
      F4(I) = DER(I)
C
C      COMPUTE DELTA Y
      DELTY(I) = SPEC*(F1(I)*CAPF1(I)+(-3.0*(F1(I)+P1(I)*VAR(I1))+2.0*
*      (F2(I)+P1(I)*Y2(I)) + 2.0*(F3(I)+P1(I)*Y3(I)) - F4(I) - P1(I)*
*      Y4(I))*CAPF2(I) + 4.0*((F1(I)+P1(I)*VAR(I1)) - (F2(I)+P1(I)*
*      Y2(I)) - (F3(I)+P1(I)*Y3(I)) + (F4(I)+P1(I)*Y4(I)))*CAPF3(I))
C
C      COMPUTE Y + DELTA Y
20   CUVAR(I1) = VAR(I1) + DELTY(I)
C      CALL CHECK FOR DECISION TO ACCEPT OR RECOMPUTE INTERVAL
      CALL CHECK
      IF (ELB) 21,21,23
C
C      UPDATE Y VALUES
21   N1 = N + 1
      DO 22 I = 2,N1
      I1 = I - 1
22   VAR(I) = VAR(I) + DELTY(I1)
      VAR(1) = VAR(1) + SPEC
C      RETURN TO COMPUTE DERIVATIVES AT Y + DELTA Y
      GO TO 2
C      RETURN TO RECOMPUTE INTERVAL
23   GO TO 5
      END
C
C-----
C
C      SUBROUTINE FTLUP (X,Y,M,N,VARI,VARD)
C
C      THIS SUBROUTINE IS A MODIFICATION OF LIBRARY INTERPOLATION
C      SUBROUTINE FTLUP   REVISED 7-7-69
C
C      IMPLICIT REAL*8(A-H,O-Z)
C      DIMENSION VARI(1),VARD(1),V(3),YY(2),II(100)
C      INITIALIZE ALL INTERVAL POINTERS TO -1.0 FOR MONOTONICITY CHECK
4    DO 4 J = 1,100
      II(J) = -1
      MA = IABS(M)
C      ASSIGN INTERVAL POINTER FOR GIVEN VARI TABLE ; THE SAME POINTER
C      WILL BE USED ON A GIVEN VARI TABLE EVERY TIME
      LOCF = VARI(1)
      LI = MOD(LOCF,100) + 1
      I = II(LI)
      IF (I.GE.0) GO TO 10
      IF (N.LT.2) GO TO 10

```



```

C                                     MONOTONICITY CHECK
      IF (VARI(2) - VARI(1)) 1,1,3
C                                     ERROR IN MONOTONICITY
      2 K = LOCF
      WRITE(*,102) J,K
102  FORMAT(1X,'TABLE BELOW OUT OF ORDER FOR FTLUP AT POSITION ',
      * I5,/,1X,'X TABLE IS STORED IN LOCATION',I5//)
      DO 103 J = 1,N
103  WRITE(*,*) VARI(J),VARD(J)
      STOP
C                                     MONOTONIC DECREASING
      1 DO 5 J = 2,N
      IF (VARI(J) - VARI(J-1)) 5,2,2
      5 CONTINUE
      GO TO 10
C                                     MONOTONIC INCREASING
      3 DO 6 J = 2,N
      IF (VARI(J) - VARI(J-1)) 2,2,6
      6 CONTINUE
C                                     INTERPOLATION
10  IF (I.LE.0) I = 1
      IF (I.GE.N) I = N - 1
      IF (N.LE.1) GO TO 8
      IF (MA.NE.0) GO TO 99
C                                     ZERO ORDER
      8 Y = VARD(1)
      GO TO 800
C                                     LOCATE I INTERVAL (X(I).LE.X.LT.X(I+1))
99  IF ((VARI(I) - X)*(VARI(I+1) - X)) 61,61,40
C                                     IN GIVES DIRECTION FOR SEARCH INTERVALS
40  SIGN1 = 1.0
      IN = SIGN(SIGN1,(VARI(I+1) - VARI(I))*(X - VARI(I)))
C                                     IF X OUTSIDE ENDPOINTS, EXTRAPOLATE FROM END INTERVAL
41  IF ((I+IN).LE.0) GO TO 61
      IF ((I+IN).GE.N) GO TO 61
      I = I + IN
      IF ((VARI(I) - X)*(VARI(I+1) - X)) 61,61,41
61  IF (MA.EQ.2) GO TO 200
C                                     FIRST ORDER
      Y = (VARD(I)*(VARI(I+1) - X) - VARD(I+1)*(VARI(I) - X))/
      * (VARI(I+1) - VARI(I))
      GO TO 800
C                                     SECOND ORDER
200 IF (N.EQ.2) GO TO 2
      IF (I.EQ.(N-1)) GO TO 209
      IF (I.EQ.1) GO TO 201
C                                     PICK THIRD POINT
      SK = VARI(I+1) - VARI(I)
      IF ((SK*(X - VARI(I-1))).LT.(SK*(VARI(I+2) - X))) GO TO 209
201 L = I
      GO TO 702
209 L = I - 1
702 V(1) = VARI(L) - X
      V(2) = VARI(L+1) - X
      V(3) = VARI(L+2) - X
      YY(1) = (VARD(L)*V(2)-VARD(L+1)*V(1))/(VARI(L+1)-VARI(L))
      YY(2) = (VARD(L+1)*V(3)-VARD(L+2)*V(2))/(VARI(L+2)-VARI(L+1))
      Y = (YY(1)*V(3) - YY(2)*V(1))/(VARI(L+2) - VARI(L))
800 II(LI) = I
      RETURN

```

END

C

C-----

C

SUBROUTINE FOFTS23 (MAXI,TOL1,TOL2,ICODE,TSG,IQ1,IQ2,IQ3)

C

C

CALLED BY MAIN TO EVALUATE TS BY NEWTON ITERATION TECHNIQUE

C

IMPLICIT REAL*8(A-H,O-Z)

COMMON /A1/ M,IX,IMAX,IPSI,MODEL,TE1,TE2

COMMON /A2/ P(200),DPDX(200),VARI(200)

COMMON /A3/ EVIINF(25),THETAI(25),MUI(25),FI(25),DELHI(25),

* CIINF(25),LI(25)

COMMON /A4/ R,MUINF,DELTA,LAMSQ,OMEGSQ,OMEGA,MU,T,U,TINF

COMMON /A5/ EPSIIL(20,25),GIL(20,25),CID(200,4)

COMMON /A6/ VAR(52),CUVAR(52),DER(51)

DIMENSION SUMGM(25),EISM(25),DEISM(25)

REAL*8 MUI,LAMSQ,MUINF,MU

C

W = 1.0

DO 1 I = 1,IMAX

1 CID(IX,I) = CIINF(I)

ITER = 0

2 ITER = ITER + 1

IF (ITER.EQ.1) THEN

ES = DELTA

PS = DSQRT(OMEGSQ - 2.*LAMSQ*(DELTA - ES))

ENDIF

IF (IQ1.EQ.1) THEN

TVIB = TINF

ELSE

TVIB = TSG

ENDIF

IF (IQ2.EQ.1) THEN

TEL = TINF

ELSE

TEL = TSG

ENDIF

C

ES = 0.0

DES = 0.0

DO 25 I = 1,IMAX

IF (M.EQ.1) EVIS = EVIINF(I)

IF (M.EQ.1) GO TO 5

TEM = DEXP(THETAI(I)/TVIB)

EVIS = (R*THETAI(I))/(MUI(I)*(TEM - 1.0))*FI(I)

DEVIS = (R*FI(I)*THETAI(I)**2)/(MUI(I)*TVIB**2)*(TEM/(TEM-1.0)**2)

5 IF (I.EQ.1) QVN = 1.0/(1.0 - DEXP(-THETAI(I)/TVIB))

IF (I.EQ.2) QVO = 1.0/(1.0 - DEXP(-THETAI(I)/TVIB))

SUMGM(I) = 0.0

SUMGE = 0.0

DSUMGE = 0.0

LII = LI(I)

C

DO 10 L = 1,LII

TEM1 = DEXP(-EPSIIL(L,I)/TEL)

SUMGM(I) = SUMGM(I) + TEM1*GIL(L,I)

SUMGE = SUMGE + TEM1*GIL(L,I)*EPSIIL(L,I)

10 DSUMGE = DSUMGE + TEM1*GIL(L,I)*EPSIIL(L,I)**2

QROTN = TSG/5.8

```

QROTO - TSG/4.2
EEIS - (R/MUI(I))*(SUMGE/SUMGM(I))
DEEIS - R/MUI(I)*(((SUMGM(I)*DSUMGE/TEL**2) - (SUMGE/TEL)**2)
* /SUMGM(I)**2)
EISM(I) - (1.5*R*TSG)/MUI(I) + (FI(I)*R*TSG)/MUI(I) + EVIS + EEIS
* + DELHI(I)/MUI(I)
DEISM(I) - 1.5*R/MUI(I) + FI(I)*R/MUI(I) + DEVIS + DEEIS
25 CONTINUE
C BETA FOR OXYGEN
IF (IQ3.EQ.2) THEN
  TD - 59500.0
  F1 - CIINF(2)*MUINF/MUI(2)
  C1A - TD/TSG
  C1A - DEXP(C1A/2.0)
  C1B - 8.8568195*(1. - CIINF(1))*C1A*(1./(TSG**2.5))
  C1 - C1A*2.0*MUINF*PS*C1B*QVO*QROTO*(SUMGM(2)/(SUMGM(4)**2))/R
  BETAO - ((F1-1.) + DSQRT((1.-F1)**2 + 4.*(F1+C1)))/(2.*(F1+C1))
  BETAN - 0.0
ENDIF
C BETA FOR NITROGEN AND OXYGEN
IF (IQ3.EQ.3) THEN
  BETAN - .1
  BETAO - 1.0
  TDN - 113500.0
  TDO - 59500.0
  F1A - 8.8568195*(1. - CIINF(1))*DEXP(TDO/TSG)*(1./(TSG**2.5))
  F2A - 12.36681*(1. - CIINF(2))*DEXP(TDN/TSG)*(1./(TSG**2.5))
  F1 - 2.0*MUINF*PS*F1A*QVO*QROTO*(SUMGM(2)/(SUMGM(4)**2))/R
  F2 - 2.0*MUINF*PS*F2A*QVN*QROTN*(SUMGM(1)/(SUMGM(3)**2))/R
30 A - ((1. - BETAO**2) + BETAN*(1. - BETAO))/(BETAO**2) - F1
  B - ((1. - BETAN**2) + BETAO*(1. - BETAN))/(BETAN**2) - F2
  ABO - (BETAN*BETAO - 2.*(1. + BETAN))/(BETAO**3)
  ABN - (1. - BETAO)/(BETAO**2)
  BBO - (1. - BETAN)/(BETAN**2)
  BBN - (BETAN*BETAO - 2.*(1. + BETAO))/(BETAN**3)
  DEN - ABO*BBN - BBO*ABN
  DBO - (-A*BBN + B*ABN)/DEN
  DBN - (-B*ABO + A*BBO)/DEN
  BETAN - BETAN + DBN
  BETAO - BETAO + DBO
  IF ((DABS(DBO).GT..00001).OR.(DABS(DBN).GT..00001)) GO TO 30
ENDIF
C DISSOCIATING CONCENTRATIONS ACROSS THE SHOCK
CID(IX,2) - (1.0 - BETAO)*CIINF(2)
CID(IX,4) - BETAO*CIINF(2)
CID(IX,1) - (1.0 - BETAN)*CIINF(1)
CID(IX,3) - BETAN*CIINF(1)
C
DO 35 I = 1,IMAX
  ES - ES + EISM(I)*CID(IX,I)
35 DES - DES + DEISM(I)*CID(IX,I)
  FAC - DSQRT(OMEGSQ - 2.0*LAMSQ*(DELTA - ES))
  FIN - (2.0*MUINF*(DELTA - ES)*FAC)/(R*(OMEGA + FAC))
  FTS - TSG - FIN
  FAC1 - R*(OMEGA+FAC)*(MUINF*(DELTA-ES)*(2.0/FAC)*LAMSQ*DES -
* 2.0*FAC*MUINF*DES) - 2.0*MUINF*(DELTA-ES)*R*LAMSQ*DES
  FAC2 - R**2*(OMEGA + FAC)**2
  DFTS - 1.0 - (FAC1/FAC2)
  PS - FAC
  IF (DFTS.EQ.0.0) THEN

```

```

      ICODE = 2
      RETURN
    ENDIF
    TSG1 = TSG - W*(FTS/DFTS)
    DELTS = DABS(TSG1 - TSG)
    TSG = TSG1
    IF (ITER.GE.MAXI) THEN
      ICODE = 1
      RETURN
    ENDIF
    IF (DELTS.GT.TOL1) THEN
      ITER1 = ITER
      GO TO 2
    ENDIF
    ITER2=ITER1+10
    IF(DELTS.LE.TOL2.OR.ITER.GE.ITER2)RETURN
    GO TO 2
  END

```

STATEMENT LISTING OF PROGRAM RADNW1.FOR

C PROGRAM FOR COMPUTING THE RADIATIVE HEAT TRANSFER AT THE BODY SURFACE
 C IN A HYPERSONIC FLOWFIELD. THIS PROGRAM IS USED IN CONJUNCTION WITH
 C AFE2.FOR
 C THIS IS RADMTE.FOR
 C RADIATION MODELS

- 1) OLSTAD MODEL
- 2) CARLSON MODEL

C
 C

```

  IMPLICIT REAL*8(A-H,O-Z)
  COMMON /A9/ MUI(25),XPST(30)
  COMMON /RAD/ NXCON,NPTS(30),TSTAG,LI(25),IMAX
  COMMON /RAD1/ RRHO(30,30),Y(30,30),YBDY(30)
  COMMON /RAD2/ RCON(30,10,30)
  COMMON /RAD3/ B(30,8,30)
  COMMON /RAD5/ OPTL(30,8,30)
  COMMON /RAD6/ ALPN(30,30),ALPO(30,30),BETN(30,30),BETO(30,30)
  COMMON /RAD7/ TEMPO(30,30),TEMP(30,30)
  COMMON /RAD8/ KAP(30,4,30)
  COMMON /RAD9/ KAPPA(30,8,30),KA(30,4,30)
  COMMON /RAD10/ EPSIIL(20,25),GIL(20,25)
  REAL OPTL,XPST,Y,YBDY,TEMPO,TEMP,Z
  REAL ALPN,ALPO,B,BETN,BETO
  REAL*8 KAPPA,KAP,KA,MUI,EPSIIL,GIL
  CHARACTER*30 UTFIL

```

C
 C

OPEN RADIATION INPUT FILES (AFE2.FOR)

```

  OPEN (UNIT=15,FILE='RADIN1.DAT',STATUS='OLD')
  OPEN (UNIT=16,FILE='RADIN2.DAT',STATUS='OLD')
  OPEN (UNIT=17,FILE='RADIN3.DAT',STATUS='OLD')

```

C

READ INPUT

```

  READ(15,1) UTFIL
1  FORMAT(A30)
  READ(15,*) MODEL,IQ3,VINF,TINF,PINF
  READ(15,*) NXPST,NXCON,IMAX,IQ4,IQ5,IQ6,IQ8,IQ9,IQ10
  IF (NXCON.GT.30) THEN
    WRITE(*,2) NXCON
2  FORMAT(1X,'NXCON = ',I3,' MAX = 30 STRLINES IN RAD.FOR')
    STOP
  ENDIF
  IF (IQ4.EQ.0) GO TO 60
  OPEN (UNIT=8,FILE=UTFIL,STATUS='UNKNOWN')
  DO 5 I = 1,IMAX
5  READ(15,*) MUI(I)
  DO 7 I = 1,IMAX
  READ(15,*) LII
  LI(I) = LII
7  READ(15,*) (GIL(L,I),L = 1,LII)
  DO 9 I = 1,IMAX
  READ(15,*) LII
9  READ(15,*) (EPSIIL(L,I),L = 1,LII)
  DO 10 I = 1,NXPST
10 READ(15,*) NPTS(I),XPST(I)
  READ(15,*) TSTAG
  READ(15,*) KSTAG
  DO 15 I = 1,KSTAG
  READ(16,*) LPS2,JJJ
  IF (IMAX.EQ.7) THEN
    READ(16,*) RCON(LPS2,1,JJJ),RCON(LPS2,2,JJJ),RCON(LPS2,3,JJJ),
    * RCON(LPS2,4,JJJ),RCON(LPS2,5,JJJ),RCON(LPS2,6,JJJ),
    * RCON(LPS2,7,JJJ),RRHO(LPS2,JJJ),TEMP(LPS2,JJJ),
    * TEMPO(LPS2,JJJ)
  ENDIF

```

```

      IF (IMAX.EQ.10) THEN
        READ(16,*) RCON(LPS2,1,JJJ),RCON(LPS2,2,JJJ),RCON(LPS2,3,JJJ),
        *          RCON(LPS2,4,JJJ),RCON(LPS2,5,JJJ),RCON(LPS2,6,JJJ),
        *          RCON(LPS2,7,JJJ),RCON(LPS2,8,JJJ),RCON(LPS2,9,JJJ),
        *          RCON(LPS2,10,JJJ),RRHO(LPS2,JJJ),TEMP(LPS2,JJJ),
        *          TEMPO(LPS2,JJJ)
      ENDIF
15  CONTINUE
19  READ(17,*,END=21) IBOB,IK
    READ(17,*) YBDY(IBOB)
    DO 20 J = 2,IK
20  READ(17,*) Y(IBOB,J-1)
    GO TO 19

```

C

INITIAL CONDITIONS

```

21  WRITE(*,22) VINFINF,PINF,TINF
    WRITE(8,22) VINFINF,PINF,TINF
22  FORMAT(1X,'FREESTREAM QUANTITIES : ',//,1X,'VINFINF = ',E12.5,1X,
    *      'cm/sec',/,1X,'PINF = ',E12.5,1X,'dynes/cm^2',/,1X,
    *      'TINF = ',E12.5,1X,'K'///)
    WRITE(*,23) MODEL
    WRITE(8,23) MODEL
23  FORMAT(1X,'VIBRATION-DISSOCIATION COUPLING MODELS',//,12X,
    *      'TYPE',11X,'NO.',//,9X,'VIB. EQUIL.',8X,'0',/,12X,'CVD',13X,
    *      '1',/,12X,'CVDV',12X,'2',/,6X,'CVDV-Preferential',5X,'3',/,
    *      12X,'PARK',12X,'4', ' = ',I3///)
    WRITE(*,24) IQ3
    WRITE(8,24) IQ3
24  FORMAT(7X,'SHOCK JUMP CONDITION MODELS',//,12X,
    *      'TYPE',11X,'NO.',//,6X,'CHEMISTRY FROZEN',6X,'1',/,5X,
    *      'N2 FROZEN, O2 DISS.',4X,'2',/,7X,'N2 AND O2 DISS.',6X,
    *      '3', ' = ',I3///)

```

C

RADIATION MODELS

```

      IF (IQ4.EQ.1) CALL OLRAD (IQ5,IQ6,IQ8,IQ9,IQ10)
C      IF (IQ4.EQ.2) CALL CARLRAD (IQ4,IQ5,IQ9)
      IF ((IQ4.EQ.3).OR.(IQ4.EQ.4)) THEN
C        CALL OLRAD (IQ5,IQ6,IQ8,IQ9,IQ10)
C        CALL CARLRAD (IQ4,IQ5,IQ9)
      ENDIF
      IF (IQ4.EQ.4) THEN
C      IF (IQ5.EQ.0) THEN
        DO 1920 K=3, NXCON
          WRITE(8,30) XPST(K)
          DO 1920 J=1,NPTS(K)
            WRITE(8,1921) J,ALPN(K,J),ALPO(K,J),BETN(K,J),BETO(K,J)
1921  FORMAT(' J=',I2,' An=',D11.4,' Ao=',D11.4/
          *      ',2X,' BN=',D11.4,' BO=',D11.4)
1920  CONTINUE
C      ENDIF
        DO 40 K = 3,NXCON
          WRITE(*,30) XPST(K)
          WRITE(8,30) XPST(K)
30  FORMAT(///,'AT X = ',D11.4,///)
          DO 40 J = 1,NPTS(K)
            WRITE(*,35) J,TEMP(K,J),RRHO(K,J),TEMPO(K,J)
            WRITE(8,35) J,TEMP(K,J),RRHO(K,J),TEMPO(K,J)
35  FORMAT(' J = ',I2,' TVN2 = ',F10.2,' RHO = ',D11.4,' T=',F10.2)
40  CONTINUE
        DO 55 K = 3,NXCON
          WRITE(8,30) XPST(K)
          DO 55 J = 1,NPTS(K)

```

```

      WRITE(8,45) J
45  FORMAT(/,' AT J = ',I2,' THE NUMBER DENSITIES ARE ',/)
      DO 55 I = 1,IMAX
      WRITE(8,50) I,RCON(K,I,J)
50  FORMAT(' SPECIES = ',I2,' N = ',D11.4)
55  CONTINUE
      ENDIF
      CLOSE(8)
60  CLOSE(15)
      CLOSE(16)
      CLOSE(17)
      STOP
      END

```

C
C-----
C

```

SUBROUTINE OLRAD (IQ5,IQ6,IQ8,IQ9,IQ10)
IMPLICIT REAL*8(A-H,O-Z)
COMMON /A9/ MUI(25),XPST(30)
COMMON /RAD/ NXCON,NPTS(30),TSTAG,LI(25),IMAX
COMMON /RAD1/ RRHO(30,30),Y(30,30),YBDY(30)
COMMON /RAD2/ RCON(30,10,30)
COMMON /RAD3/ B(30,8,30)
COMMON /RAD5/ OPTL(30,8,30)
COMMON /RAD6/ ALPN(30,30),ALPO(30,30),BETN(30,30),BETO(30,30)
COMMON /RAD7/ TEMPO(30,30),TEMP(30,30)
COMMON /RAD8/ KAP(30,4,30)
COMMON /RAD9/ KAPPA(30,8,30),KA(30,4,30)
COMMON /RAD10/ EPSIIL(20,25),GIL(20,25)
C  REAL OPTL,XPST,Y,YBDY,TEMPO,TEMP,Z
C  REAL ALPN,ALPO,B,BETN,BETO,EPSIIL,GIL
REAL*8 K11,K22,N,NN,KAPPA,KAP,KA,MUI
REAL*8 A(4,5)

```

C

```

      WRITE(8,1)
1  FORMAT(///' OLSTAD RADIATION MODEL')
      IF(IQ8.EQ.0) THEN
      IF(IQ6.EQ.0) WRITE(8,2)
      IF(IQ6.EQ.1) WRITE(8,3)
      ENDIF
2  FORMAT('/' BETA=BETA(Te)')
3  FORMAT('/' BETA=BETA(Tt)')

```

C

NUMBER DENSITIES

```

      DO 12 K=3,NXCON
      DO 12 J=1,NPTS(K)
      DO 10 II=1,IMAX
      XXX=RRHO(K,J)*6.023D+23
      XXX=XXX*RCON(K,II,J)/MUI(II)
10  RCON(K,II,J)=XXX
12  CONTINUE
      IF(IQ10.EQ.0)GO TO 15

```

C

ELECTRON TEMPERATURE

```

      DO 14 K=3,NXCON
      NUP=NPTS(K)-1
      DO 14 J=1,NUP
      T=TEMPO(K,J)
      FNA=RCON(K,3,J)
      FNI=RCON(K,9,J)
      FNE=RCON(K,7,J)
      FNM=RCON(K,1,J)

```



```
FOA-RCON(K,4,J)
FOI-RCON(K,10,J)
ITRTE=0
FOM-RCON(K,2,J)
EI-2.3322D-11
FAV-6.023D23
EO-2.173D-11
SA-1.52D-15
F1=0
F2=0
F0=0
574 ITRTE=ITRTE+1
IF(ITRTE.GT.1)GO TO 571
KK=1
569 KK=KK+1
IF(KK.GT.50)TEMP(K,J)=2000
IF(KK.GT.50)GO TO 14
TE=1000*KK
571 SM=0.5355D-19*TE+.696D-15
IF(ITRTE.GT.50)GO TO 573
SI=1.53614D08*(TE**3)/FNE
SI=DSQRT(SI)
SI=(4.38384D-06/(TE**2))*DLOG(SI)
FKF=4.16D13*(TE**( 0.5D00))*DEXP(-1.20D05/TE)
FK1=5.49D13*(TE**( 0.5D00))*DEXP(-1.045D05/TE)
FKB=2.3D21*(TE**(-1.0D00))*DEXP(4.9D4/TE)
FK5=1.53D22*(TE**(-1.0D00))*DEXP(5.3D4/TE)
WEA=FKF*FNA-FKB*(FNI/FAV)*FNE
W2E=FK1*FOA-FK5*(FOI/FAV)*FNE
SX=(FNA+FOA)*SA+(FNI+FOI)*SI+(FNM+FOM)/2.0D0*SM
W1=WEA+W2E
FE=T-1.23357D-10/SX/DSQRT(TE)*(WEA*EI+W2E*EO+W1*3.45D-16*TE)-TE
IF(ITRTE.GT.1)GO TO 568
IF((FE/DABS(FE)*F0).LT.0.D00)GO TO 567
F0=FE
T0=TE
GO TO 569
567 F1=FE
T1=TE
566 TE=(T0*F1-T1*F0)/(F1-F0)
IF(DABS(TE-T1).LT.1.D00)GO TO 573
IF(DABS(TE-T0).LT.1.D00)GO TO 573
GO TO 574
568 IF((FE/DABS(FE)*F0/DABS(F0)).LT.0.0D00)GO TO 564
T0=TE
F0=FE
GO TO 566
564 F1=FE
T1=TE
GO TO 566
573 IF(TE.GT.T)TE=T
IF(ITRTE.GT.50)WRITE(*,572)
IF(ITRTE.GT.50)WRITE(8,572)
572 FORMAT(' ','TE DID NOT CONVERGE')
TEMP(K,J)=TE
14 CONTINUE
15 CONTINUE
```

NONEQUILIBRIUM RADIATION CORRECTION

```

DO 122 J=1,NPTS(K)
IF(J.EQ.NPTS(K)) THEN
  ALPN(K,J)=0.D+00
  ALPO(K,J)=0.D+00
  BETO(K,J)=0.D+00
  BETN(K,J)=0.D+00
  GO TO 122
ENDIF
Q1N=0.D+0
Q2N=0.D+0
Q3N=0.D+0
DO 112 III=1,LI(3)
112 Q1N=Q1N+GIL(III,3)*DEXP(-EPSIIL(III,3)/TEMP(K,J))
DO 113 III=1,LI(9)
113 Q2N=Q2N+GIL(III,9)*DEXP(-EPSIIL(III,9)/TEMP(K,J))
DO 130 III=1,LI(1)
130 Q3N=Q3N+GIL(III,1)*DEXP(-EPSIIL(III,1)/TEMP(K,J))
RHO1=RCON(K,3,J)*1.401D+01/6.023D+23
RHO2=RCON(K,9,J)*1.401D+01/6.023D+23
RHO3=RCON(K,1,J)*2.802D+01/6.023D+23
RHO4N=RHO1+RHO2+RHO3
QVIBN=1.D+0/(1.D+0-DEXP(-3.39D+03/TEMP(K,J)))
QROTN=TEMP(K,J)/5.8D+0
Q1O=0.D+0
Q2O=0.D+0
Q3O=0.D+0
DO 152 III=1,LI(4)
152 Q1O=Q1O+GIL(III,4)*EXP(-EPSIIL(III,4)/TEMP(K,J))
DO 153 III=1,LI(10)
153 Q2O=Q2O+GIL(III,10)*DEXP(-EPSIIL(III,10)/TEMP(K,J))
DO 160 III=1,LI(2)
160 Q3O=Q3O+GIL(III,2)*DEXP(-EPSIIL(III,2)/TEMP(K,J))
RHO1=RCON(K,4,J)*1.600D+01/6.023D+23
RHO2=RCON(K,10,J)*1.600D+01/6.023D+23
RHO3=RCON(K,2,J)*3.200D+01/6.023D+23
RHO4O=RHO1+RHO2+RHO3
QVIBO=1.D+0/(1.D+0-DEXP(-2.27D+03/TEMP(K,J)))
QROTO=TEMP(K,J)/4.2D+0
C ALPHAN=0.5D+0
C BETAN=0.5D+0
C ALPHAO=0.5D+0
C BETAO=0.5D+0
C1=2.4701D01*RHO4N*QVIBN*QROTN*Q3N*DEXP(1.135D+05/TEMP(K,J))
C1=C1/(Q1N**2*TEMP(K,J)**1.5)
C2=1.7725D+01*RHO4O*QVIBO*QROTO*Q3O*DEXP(5.950D+04/TEMP(K,J))
C2=C2/(Q1O**2*TEMP(K,J)**1.5)
C C3=RHO4N*8.909D+06*Q1N*DEXP(1.69D+05/TEMP(K,J))
C C3=C3/(Q2N*TEMP(K,J)**1.5)
C C4=C3*RHO4O/RHO4N*0.875625D+0
C C5=RHO4O*7.801D+06*Q1O*DEXP(1.58D+05/TEMP(K,J))
C C5=C5/(Q2O*TEMP(K,J)**1.5)
C C6=C5*RHO4N/RHO4O*1.142D+0
C ITAB=0
C 132 ITAB=ITAB+1
C IF(ITAB.LT.51)GO TO 133
C WRITE(*,1331)K,J
C WRITE(8,1331)K,J
C1331 FORMAT(' ','CORR FAC FAILED TO CONVERGE',2I5)
C GO TO 127
C 133 A(1,5)=- (BETAN**2*(1.D+0-ALPHAN)**2*C1+BETAN-1.D+0)

```

```

C      A(3,5)--(BETAO**2*(1.D+0-ALPHAO)**2*C2+BETAO-1.D+0)
C      A(2,5)--(BETAN*ALPHAN**2*C3+C4*ALPHAN*BETAO*ALPHAO+ALPHAN-1.D+0)
C      A(4,5)--(C5*BETAO*ALPHAO**2+C6*BETAN*ALPHAN*ALPHAO+ALPHAO-1.D+0)
C      A(1,1)--(2.D+0)*(1.D+0-ALPHAN)*BETAN**2
C      A(1,2)--(2.D+0)*BETAN*(1.D+0-ALPHAN)**2+1.D+0
C      A(1,3)=0.0D+00.
C      A(1,4)=0.0D+00
C      A(2,1)--(2.D+0)*C3*ALPHAN*BETAN+C4*BETAO*ALPHAO+1.D+0
C      A(2,2)=C3*ALPHAN**2
C      A(2,3)=C4*ALPHAN*BETAO
C      A(2,4)=C4*ALPHAN*ALPHAO
C      A(3,1)=0.D+00
C      A(3,2)=0.D+00
C      A(3,3)--(2.D+0)*(1.D+0-ALPHAO)*BETAO**2
C      A(3,4)--(2.D+0)*BETAO*(1.D+0-ALPHAO)**2+1.D+0
C      A(4,1)=C6*BETAN*ALPHAO
C      A(4,2)=C6*ALPHAN*ALPHAO
C      A(4,3)--(2.D+0)*C5*BETAO*ALPHAO+C6*BETAN*ALPHAN+1.D+0
C      A(4,4)=C5*ALPHAO**2
C      DO 121 I=1,4
C      II=I+1
C      DO 125 JJ=II,5
C 125  A(I,JJ)=A(I,JJ)/A(I,I)
C      DO 123 KK=1,4
C      IF(KK.EQ.I)GO TO 123
C      DO 124 JJ=II,5
C 124  A(KK,JJ)=A(KK,JJ)-A(KK,I)*A(I,JJ)
C 123  CONTINUE
C 121  CONTINUE
C      ALPHAN=ALPHAN+A(1,5)
C      BETAN=BETAN+A(2,5)
C      ALPHAO=ALPHAO+A(3,5)
C      BETAO=BETAO+A(4,5)
C      DO 126 I=1,4
C      IF (DABS(A(I,5)).GT.1.D-05)GO TO 132
C 126  CONTINUE
C 127  ALPHN=RCON(K,9,J)/(RCON(K,9,J)+RCON(K,3,J))
C      BET1=(RCON(K,9,J)+RCON(K,3,J))/(2.D+0*RCON(K,1,J)+
C      *      RCON(K,9,J)+RCON(K,3,J))
C      WRITE(8,1271)K,J,ALPHAN,BETAN,ALPHN,BET1
C1271  FORMAT(' ',2I5,4E12.4)
C      ALPHO=RCON(K,10,J)/(RCON(K,10,J)+RCON(K,4,J))
C      BET2=(RCON(K,10,J)+RCON(K,4,J))/(2.D+0*RCON(K,2,J)+
C      *      RCON(K,10,J)+RCON(K,4,J))
C      WRITE(8,1271)K,J,ALPHAO,BETAO,ALPHO,BET2
C      BETN(K,J)=(BET1**2/(1.D+0-BET1))*((1.D+0-BETAN)/(BETAN**2)
C      *      *((1.D+0-ALPHN)**2/(1.D+0-ALPHAN)**2)
C      BETN(K,J)=(RCON(K,3,J)/RCON(K,1,J))*RCON(K,3,J)*C1*1.162D-23
C      *      /RHO4N
C      C7=RHO40/RHO4N*0.875625D+0
C      C8=BET2/BET1*ALPHO/ALPHN
C      C9=BETAO/BETAN*ALPHAO/ALPHAN
C1888  ALPN(K,J)=BET1*(ALPHN**2/(1.D+0-ALPHN))
C      *      *((1.D+0-ALPHAN)/(BETAN*ALPHAN**2))
C      *      *(1.D+0+C7*C8)/(1.D+0+C7*C9)
C      ALPN(K,J)=RCON(K,9,J)/RCON(K,3,J)*RCON(K,7,J)*Q1N/Q2N*
C      *      DEXP(1.69D+05/TEMP(K,J))/(4.826D+15*TEMP(K,J)**1.5)
C      BETO(K,J)=(BET2**2/(1.D+0-BET2))*((1.D+0-BETAO)/(BETAO**2)
C      *      *((1.D+0-ALPHO)**2/(1.D+0-ALPHAO)**2)
C      BETO(K,J)=(RCON(K,4,J)/RCON(K,2,J))*RCON(K,4,J)*C2*1.328D-23

```

```

*          /RHO40
C1999 ALPO(K,J)=BET2*(ALPHO**2/(1.D+0-ALPHO))
C      *          *((1.D+0-ALPHAO)/(BETAO*ALPHAO**2))
C      *          *(1.D+0+1.D+0/C7/C8)/(1.D+0+1.D+0/C7/C9)
      ALPO(K,J)=RCON(K,10,J)/RCON(K,4,J)*RCON(K,7,J)*Q10/Q20*
      *          DEXP(1.575D+05/TEMP(K,J))/(4.826D+15*TEMP(K,J)**1.5)
122  CONTINUE
C      ENDIF
C
C          ABSORPTION COEFFICIENTS
      DO 20 K=3,NXCON
      DO 20 J=1,NPTS(K)
      TT=TEMP(K,J)/168800.0
      IF(IQ6.EQ.1) THEN
          TM=TEMPO(K,J)/168800.0
      ELSE
          TM=TT
      ENDIF
C      IF((IQ6.EQ.1).AND.(TEMPO(K,J).LE.8.D+03)) GO TO 20
C      IF(TEMP(K,J).LE.8.D+03) GO TO 20
      IF(IQ8.EQ.0) THEN
          O=BETO(K,J)
          N=BETN(K,J)
      ELSE
          O=1.D+0
          N=1.D+0
      ENDIF
      KAPPA(K,4,J)=5.D-19*O*RCON(K,2,J)+5.D-20*N*RCON(K,1,J)
      KAP(K,4,J)=1.7D-17*RCON(K,3,J)*DEXP(-.246/TT)
      KA(K,4,J)=0.D+0
      KAPPA(K,3,J)=2.D-18*(N*RCON(K,1,J)+O*RCON(K,2,J))+KAPPA(K,4,J)
      KAP(K,3,J)=2.1D-17*RCON(K,3,J)*DEXP(-.165/TT)+KAP(K,4,J)
      KA(K,3,J)=0.D+0
      KAPPA(K,2,J)=5.1D-18*(N*RCON(K,1,J)+O*RCON(K,2,J))+KAPPA(K,3,J)
      KAP(K,2,J)=KAP(K,3,J)
      KA(K,2,J)=5.1D-18*RCON(K,4,J)
      KAPPA(K,1,J)=2.D-17*O*RCON(K,2,J)+4.D-16*N*RCON(K,1,J)
C      +KAPPA(K,2,J)
      KAP(K,1,J)=1.1D-17*RCON(K,3,J)+KAP(K,2,J)
      KA(K,1,J)=KA(K,2,J)
      IF(IQ5.EQ.0) THEN
          N=ALPN(K,J)
          O=ALPO(K,J)
      ELSE
          N=1.D+0
          O=1.D+0
      ALPN(K,J)=N
      ALPO(K,J)=O
      ENDIF
      IF(IQ8.EQ.0) THEN
          NN=BETN(K,J)
          OO=BETO(K,J)
      ELSE
          NN=1.D+0
          OO=1.D+0
          BETN(K,J)=NN
          BETO(K,J)=OO
      ENDIF
      KAPPA(K,5,J)=7.7D-17*(NN*RCON(K,1,J)+OO*RCON(K,2,J))
C      *DEXP(-.49/TT)+
C      2.6D-17*(RCON(K,3,J)*N+RCON(K,4,J)*O)*DEXP(-.723/TT)

```

```

      KAPPA(K,6,J)=-2.D-18*RCN(K,2,J)*OO+6.OO-18*(RCN(K,3,J)*N+
C      RCON(K,4,J)*O)*DEXP(-.379/TT)+KAPPA(K,5,J)
      IF(RCON(K,7,J).LE.1.D+OO) THEN
        KAPPA(K,7,J)=KAPPA(K,6,J)
        GO TO 909
      ENDIF
      KAPPA(K,7,J)=1.2D03*((RCN(K,3,J)*N+RCON(K,4,J)*O)/RCON(K,7,J))*
C      DEXP(-.489/TT)+KAPPA(K,6,J)
909 KAPPA(K,8,J)=3.2D-17*(RCN(K,3,J)*N+RCON(K,4,J)*O)*DEXP(-.631/TT)
C      +KAPPA(K,5,J)
C      ELSE
C      KAPPA(K,5,J)=7.7D-17*(RCN(K,1,J)+RCON(K,2,J))*DEXP(-.49/TT)+
C      C      2.6D-17*(RCN(K,3,J)+RCON(K,4,J))*DEXP(-.723/TT)
C      KAPPA(K,6,J)=2.D-18*RCN(K,2,J)+6.OO-18*(RCN(K,3,J)+RCON(K,4,J))
C      C      *DEXP(-.379/TT)+KAPPA(K,5,J)
C      IF(RCON(K,7,J).LE.1.D+OO) THEN
C      KAPPA(K,7,J)=KAPPA(K,6,J)
C      GO TO 910
C      ENDIF
C      KAPPA(K,7,J)=1.2D03*((RCN(K,3,J)+RCON(K,4,J))/RCON(K,7,J))*
C      C      DEXP(-.489/TT)+KAPPA(K,6,J)
C 910 KAPPA(K,8,J)=3.2D-17*(RCN(K,3,J)+RCON(K,4,J))*DEXP(-.631/TT)
C      C      +KAPPA(K,5,J)
C      ENDIF
20 CONTINUE

C      PLANCK FUNCTION

      DO 30 K=3,NXCON
      DO 30 J=1,NPTS(K)
      TT=TEMP(K,J)/168800.0
C      IF(TEMP(K,J).LE.8.D+03) GO TO 30
      BLACK=(TT*168800.)*4*5.6696D-12/3.1415927
      FF=15.D+00/(3.1415927*4)
      X1=1./TT
      B1=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C      DEXP(-2.*X1)*0.5*(.75+1.5*X1+1.5*X1**2+X1**3))
      X1=0.935/TT
      B2=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C      DEXP(-2.*X1)*0.5*(.75+1.5*X1+1.5*X1**2+X1**3))
      X1=0.835/TT
      B3=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C      DEXP(-2.*X1)*0.5*(.75+1.5*X1+1.5*X1**2+X1**3))
      X1=0.754/TT
      B4=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C      DEXP(-2.*X1)*0.5*(.75+1.5*X1+1.5*X1**2+X1**3))
      X1=0.473/TT
      B6=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C      DEXP(-2.*X1)*0.5*(.75+1.5*X1+1.5*X1**2+X1**3))
      X1=0.213/TT
      B8=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C      DEXP(-2.*X1)*0.5*(.75+1.5*X1+1.5*X1**2+X1**3))
      B8=1-B8
      B2=B2-B1
      B3=B3-B2
      B4=B4-B3
      B6=B6-B4
      B5=1-B8-B6-B4-B3-B2-B1
      B(K,1,J)=BLACK*B1
      B(K,2,J)=BLACK*B2
      B(K,3,J)=BLACK*B3
      B(K,4,J)=BLACK*B4

```

```

B(K,6,J)=BLACK*B6
B(K,7,J)=BLACK*2.4D-21*RCON(K,7,J)*B6*DEXP(.162/TT)
B(K,8,J)=BLACK*B8
B(K,5,J)=BLACK*B5

```

30 CONTINUE

C

TAU'S

```

DO 100 K=3,NXCON
DO 100 I=1,8
  IF(I.GT.4) THEN
    OPTL(K,I,1)=KAPPA(K,I,1)*DABS(Y(K,1)-YBDY(K))
  ELSE
    OPTL(K,I,1)=(KAPPA(K,I,1)+KAP(K,I,1)+KA(K,I,1))
    C      *DABS(Y(K,1)-YBDY(K))
  ENDIF
DO 100 J=2,NPTS(K)
  IF(I.GT.4) THEN
    K11=KAPPA(K,I,J)+KAPPA(K,I,J-1)
  ELSE
    K11=KAPPA(K,I,J)+KAP(K,I,J)+KAPPA(K,I,J-1)+KAP(K,I,J-1)
    C      +KA(K,I,J)+KA(K,I,J-1)
  ENDIF
  OPTL(K,I,J)=OPTL(K,I,J-1)+(5.D-1*(K11)*DABS(Y(K,J-1)
    C      -Y(K,J)))

```

100 CONTINUE

C

QRW

```

DO 200 K=3,NXCON
WRITE(*,102) XPST(K)
WRITE(8,102) XPST(K)
102 FORMAT(///,' AT X = ',D11.4)
WQR=0.0
DO 180 I=1,8
CALL FEI(E1I,E2I,E3I,OPTL(K,I,1))
EOLD=E2I
JIN=1
IF(I.GT.4) THEN
  K11=KAPPA(K,I,1)*B(K,I,1)
ELSE
  IF(IQ5.EQ.1) THEN
    ALPN(K,1)=1.D+0
    ALPO(K,1)=1.D+0
  ENDIF
  K11=KAPPA(K,I,1)*B(K,I,1)+KAP(K,I,1)*B(K,I,1)*ALPN(K,1)+
    C      KA(K,I,1)*B(K,I,1)*ALPO(K,1)
ENDIF
SUM=3.1415927*EOLD*K11*DABS(Y(K,1)-YBDY(K))
IF(IQ9.EQ.0) WRITE(8,2221) JIN,I,SUM
DO 150 J=2,NPTS(K)
IF(I.GT.4) THEN
  K11=KAPPA(K,I,J)*B(K,I,J)
  K22=KAPPA(K,I,J-1)*B(K,I,J-1)
ELSE
  IF(IQ5.EQ.1) THEN
    ALPN(K,J)=1.D+0
    ALPO(K,J)=1.D+0
  ENDIF
  K11=KAPPA(K,I,J)*B(K,I,J)+KAP(K,I,J)*B(K,I,J)*ALPN(K,J)+
    C      KA(K,I,J)*B(K,I,J)*ALPO(K,J)
  K22=KAPPA(K,I,J-1)*B(K,I,J-1)+
    C      KAP(K,I,J-1)*B(K,I,J-1)*ALPN(K,J-1)+
    C      KA(K,I,J-1)*B(K,I,J-1)*ALPO(K,J-1)

```

```

      ENDIF
      Z=OPTL(K,I,J)
      CALL FEI(E1I,E2I,E3I,Z)
      STSUM=3.14159*(K11*E2I+K22*EOLD)*DABS(Y(K,J-1)-Y(K,J))
      IF(IQ9.EQ.0) THEN
        WRITE(8,2221) J,I,STSUM
2221  FORMAT(1X,' AT J =',I2,' BAND ',I2,' QR =',D11.4)
      ENDIF
145  SUM=SUM+STSUM
      EOLD=E2I
150  CONTINUE
      WRITE(*,103) I,SUM
      WRITE(8,103) I,SUM
103  FORMAT(/,' FOR BAND ',I2,' QR =',D11.4,' WATTS/SQ.CM.')
      WQR=WQR+SUM
180  CONTINUE
      WRITE(*,101) WQR
      WRITE(8,101) WQR
101  FORMAT(/' TOTAL QR =',D11.4,' WATTS/SQ.CM.')
200  CONTINUE
      RETURN
      END

```

C

C-----

C

```

      SUBROUTINE FEI(E1I,E2I,E3I,Z)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /RAD1/ RRHO(30,30),Y(30,30),YBDY(30)
      COMMON /RAD3/ B(30,8,30)
      COMMON /RAD5/ OPTL(30,8,30)
C     REAL OPTL,XPST,Y,YBDY,TEMPO,TEMP,Z
C     REAL ALPN,ALPO,B,BETN,BETO
      CALL EXPI(Z,E1I,AUX)
      E2I=DEXP(-Z)-Z*E1I
      E3I=(DEXP(-Z)-Z*E2I)/2.0
C     E2I=DEXP(-Z*DSQRT(3.D+00))
      RETURN
      END

```

C

C-----

C

```

      SUBROUTINE EXPI(X,RES,AUX)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /RAD1/ RRHO(30,30),Y(30,30),YBDY(30)
      COMMON /RAD3/ B(30,8,30)
      COMMON /RAD5/ OPTL(30,8,30)
C     REAL OPTL,XPST,Y,YBDY,TEMPO,TEMP,Z
C     REAL ALPN,ALPO,B,BETN,BETO
      IF(X-1.) 2,1,1
1     YY=1./X
      AUX=1.-YY*(((YY+3.377358)*YY+2.052156)*YY+2.709479D-01)/(((YY*
C1.072553+5.716943)*YY+6.945239)*YY+2.593888)*YY+2.709496D-01)
      RES=AUX*YY*DEXP(-X)
      RETURN
2     IF(X+3.) 6,6,3
3     AUX=(((((((7.122452D-7*X-1.766345D-6)*X+2.928433D-5)*X-2.335379D-4
C)*X+1.664156D-3)*X-1.041576D-2)*X+5.555682D-2)*X-2.500001D-1)*X
C+9.999999D-1
      RES=-1.D+30
      IF(X) 4,5,4

```

```

4 RES=X*AUX-DLOG(ABS(X))-5.772157D-1
5 RETURN
6 IF(X+9.) 8,8,7
7 AUX=1.-((((5.176245D-2*X+3.061037)*X+3.243655D+1)*X+2.244234D+2)*X
C+2.486697D+2)/((((X+3.995161)*X+3.893944D+1)*X+2.263818D+1)*X
C+1.807837D+2)
GO TO 9
8 YY=9./X
AUX=1.-YY*(((YY+7.659824D-1)*YY-7.271015D-1)*YY-1.080693)/((((YY
C*2.518750+1.122927D+1)*YY+5.921405)*YY-8.666702)*YY-9.724216)
9 RES=AUX*DEXP(-X)/X
RETURN
END

```


LISTING OF INPUT FILE RR3W.NUM

0,1,0,0,0,0,0,0,0,0,1
0,0,0,2,0,0,0,0,0,0,1
0,0,0,0,1,0,0,0,0,0,1
0,0,1,1,0,0,0,0,0,0,1
1,0,0,0,0,0,0,0,0,0,1
0,0,2,0,0,0,0,0,0,0,1
0,1,1,0,0,0,0,0,0,0,0
0,0,0,1,1,0,0,0,0,0,0
1,0,0,1,0,0,0,0,0,0,0
0,0,1,0,1,0,0,0,0,0,0
0,0,0,0,0,1,1,0,0,0,0
0,0,1,1,0,0,0,0,0,0,0
0,0,2,0,0,0,0,0,0,0,0
0,0,0,0,0,0,1,1,0,0,0
0,0,2,0,0,0,0,0,0,0,0
0,0,1,0,0,0,1,0,1,0,0
0,0,1,0,0,0,0,0,1,0,0
0,0,0,0,0,0,1,0,2,0,0
0,0,1,0,0,0,1,0,0,0,0
0,0,0,0,0,0,2,0,1,0,0
0,0,0,1,0,0,1,0,0,0,0
0,0,0,0,0,0,2,0,0,1,0
0,0,0,2,0,0,0,0,0,0,0
0,1,0,0,0,0,0,0,0,0,0
0,0,1,1,0,0,0,0,0,0,0
0,0,0,0,1,0,0,0,0,0,0
0,0,2,0,0,0,0,0,0,0,0
1,0,0,0,0,0,0,0,0,0,0
0,0,0,1,1,0,0,0,0,0,0
0,1,1,0,0,0,0,0,0,0,0
0,0,1,0,1,0,0,0,0,0,0
1,0,0,1,0,0,0,0,0,0,0
0,0,1,1,0,0,0,0,0,0,0
0,0,0,0,0,1,1,0,0,0,0
0,0,0,0,0,0,1,1,0,0,0
0,0,2,0,0,0,0,0,0,0,0
0,0,1,0,0,0,1,0,1,0,0
0,0,2,0,0,0,0,0,0,0,0
0,0,0,0,0,0,1,0,2,0,0
0,0,1,0,0,0,0,0,1,0,0
0,0,0,0,0,0,2,0,1,0,0
0,0,1,0,0,0,1,0,0,0,0
0,0,0,0,0,0,2,0,0,1,0
0,0,0,1,0,0,1,0,0,0,0

608

C-7

5.675E-04,5.675E-04,5.675E-04,5.675E-04,5.675E-04,5.675E-04,5.675E-04,5.675E-04,
5.675E-04,5.675E-04
0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0
-.09952,-.09952,-.09952,-.09952,-.09952,-.09952,-.09952,-.09952,-.09952
.37177,.37177,.37177,.37177,.37177,.37177,.37177,.37177,.37177
0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0
-.2226,-.2226,-.2226,-.2226,-.2226,-.2226,-.2226,-.2226,-.2226
-.2226,-.2226,-.2226,-.2226,-.2226,-.2226,-.2226,-.2226,-.2226
0,0,0,0,0,0,0,0,0,0
-.09952,-.09952,-.09952,-.09952,-.09952,-.09952,-.09952,-.09952,-.09952
0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0
163.63,163.63,163.63,163.63,163.63,163.63,163.63,163.63,163.63,163.63
179.32,179.32,179.32,179.32,179.32,179.32,179.32,179.32,179.32,179.32
0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0
135.3,135.3,135.3,135.3,135.3,135.3,135.3,135.3,135.3,135.3
135.3,135.3,135.3,135.3,135.3,135.3,135.3,135.3,135.3,135.3
0,0,0,0,0,0,0,0,0,0
163.63,163.63,163.63,163.63,163.63,163.63,163.63,163.63,163.63,163.63
0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0
1,3,6,2
3,2,1,3,1,3
4,6,4,6
5,3,1,5,1
2,2,2,4,2,4,2,4
1,6,3
1
2,4,2,2
1,3,5,5,1,5
4,6,4,4,2
0,71592,85343,99212
0,11342,18879,51385,52104,71026
0,27659,27670,41496
0,228,326,22831,48622
0,174,63597,65381,76676,75361,87568,86359
0,57528,84205
0
0,13191,36634,92977.4
0,70.5,188.5,22036.5,47029.68,67864.8
0,38571,38601,58223.5,58225
46,18800,3393.9685,20.797195,1.0801997E-02,-7.32343E-04
33,9896,2273.1279,17.365295,7.8529954E-02,-2.057468E-03
0,0,0,0,0,0
0,0,0,0,0,0
0,0,0,0,0,0
0,0,0,0,0,0
0,0,0,0,0,0
0,0,0,0,0,0
0,0,0,0,0,0
0,0,0,0,0,0
0,0,0,0,0,0
N2
O2
N
O
NO
NO+

E-
 N2+
 N+
 O+
 $O_2 + M \rightarrow 2O + M$
 $2O + M \rightarrow O_2 + M$
 $NO + M \rightarrow N + O + M$
 $N + O + M \rightarrow NO + M$
 $N_2 + M \rightarrow 2N + M$
 $2N + M \rightarrow N_2 + M$
 $N + O_2 \rightarrow NO + O$
 $NO + O \rightarrow N + O_2$
 $N_2 + O \rightarrow NO + N$
 $NO + N \rightarrow N_2 + O$
 $NO + E \rightarrow N + O$
 $N + O \rightarrow NO + E$
 $N + N \rightarrow N_2 + E$
 $N_2 + E \rightarrow N + N$
 $N + N \rightarrow N + E + N$
 $N + E + N \rightarrow N + N$
 $N + N \rightarrow N + E + N$
 $N + E + N \rightarrow N + N$
 $N + E \rightarrow N + 2E$
 $N + 2E \rightarrow N + E$
 $O + E \rightarrow O + E + E$
 $O + E + E \rightarrow O + E$

STATEMENT LISTING OF PROGRAM AFENEW

PROGRAM AFETEW (IBM-3090, "WYLBUR" VERSION)

AN INVERSE HYPERSONIC FLOW SOLUTION FOR AN AFE/AOTV BODY IN
CHEMICAL NONEQUILIBRIUM USING COUPLED VIBRATION-DISSOCIATION MODELS,
SHOCK JUMP APPROXIMATIONS, ELECTRON TEMPERATURE MODELING, AND
UNCOUPLED RADIANT HEAT TRANSFER USING A METHOD DERIVED BY S.MASLEN

BASED ON AN ORIGINAL PROGRAM BY B.L. WEIGEL FOR WILLIAM L. GROSE
VIRGINIA POLYTECHNIC INSTITUTE

MAJOR PROGRAM REVISIONS BY G.BOB SKILL, R.GREENDYKE, AND L.CARLSON
TEXAS A&M UNIVERSITY

CALITH : INTEGRATION ROUTINE BY CHARLES E. TREANOR
THE NUMERICAL INTEGRATION ALGORITHM USED IS FOUND IN A METHOD
FOR THE NUMERICAL INTEGRATION OF COUPLED FIRST ORDER
DIFFERENTIAL EQUATIONS WITH GREATLY DIFFERENT TIME CONSTANTS

FOFE - EVALUATE E BY NEWTON ITERATION METHOD
FOFTS1 - EVALUATE TS BY NEWTON ITERATION METHOD, CHEMISTRY FROZEN
FOFT23 - FOFTS1 WITH N2 FROZEN, O2 DISSOCIATING OR BOTH N2 & O2 DISS.
FTLUP - INTERPOLATION ROUTINE

DATA FILES

UNIT 5 IS USED TO READ DATA FROM FILE SCREEN
UNIT 7 IS USED TO READ DATA FROM INPUT DATA FILE
UNIT 8 IS USED FOR THE OUTPUT DATA FILE
UNIT 9 IS USED TO STORE VIBRATIONAL ENERGY DATA AT THE SHOCK (FORMATTED)
UNIT 10 IS USED TO STORE ALL STAGNATION QUANTITIES
UNIT 11 IS USED TO STORE ALL EVIS DATA FOR COMPUTATIONAL PURPOSES
UNIT 12 IS USED TO STORE ALL PRESSURE DATA
UNIT 13 IS USED TO STORE ALL DPDX DATA
UNIT 14 IS USED TO STORE QUANTITIES FOR PHYSICAL SPACE CALCULATIONS
UNIT 15 IS USED TO STORE ALL WARNINGS DURING PROGRAM EXECUTION

INPUT-NAMELIST

DELX = INCREMENT ALONG SHOCK , cm
ZSTERM = LENGTH OF SYMMETRY AXIS , Z , cm
IMAX = MAX. NO. OF I-S SPECIES , LESS THAN OR EQUAL TO 25
JMAX = MAX. NO. OF J-S REACTIONS , LESS THAN OR EQUAL TO 50
MJ = CODE INDICATING WHICH SPECIES , I , TO USE TO CALCULATE
COUPLING FACTOR, PHI SUB J , FOR REACTION J
M = 1 FOR VIBRATIONAL NON-EQUILIBRIUM
= 0 FOR VIBRATIONAL EQUILIBRIUM
R = UNIVERSAL GAS CONSTANT , erg/(mole-K)
GAMMA = RATIO OF SPECIFIC HEATS
CIINF = FREE STREAM MASS FRACTION FOR EACH SPECIES
PINF = FREE STREAM PRESSURE , dynes/cm**2
TINF = FREE STREAM TEMPERATURE , K
VINP = FREE STREAM VELOCITY , cm/sec
MUI = MOLECULAR WT. FOR EACH SPECIES , gm/mole
THETAI = CHARACTERISTIC VIBRATIONAL TEMPERATURE , K
DGENI = FUDGE FACTOR TO PERMIT APPROXIMATING POLYATOMIC MOLECULE
BY A DIATOMIC MOLECULE
FI = 0 FOR MONATOMIC SPECIES
= 1 FOR ALL OTHERS
DELHI = HEAT OF FORMATION , ergs/mole
DELI = DISSOCIATION ENERGY OF SPECIES , K
EVI = VIBRATIONAL ENERGY OF SPECIES , ergs/g
BI = $3/2 \ln(2\pi Mi^*k/h^*2) + \ln(k) + FI(I) \ln(THETAI(Rotational))$
+ $\ln(GIL(I,1))$ FOR EACH SPECIE I
LI = NUMBER OF ELECTRONIC LEVELS FOR EACH SPECIES (LI.LE.20)
GIL = DEGENERACY OF L-TH ELECTRONIC LEVEL FOR I-TH SPECIES
EPSIIL = L-TH ELECTRONIC ENERGY LEVEL FOR I-TH SPECIES , K
AJ = FREQUENCY FACTOR IN ARRENIUS TYPE RATE EQN.
BJ = TEMPERATURE EXPONENT IN ARRENIUS TYPE EQN.
EJ = ACTIVATION ENERGY IN ARRENIUS TYPE EQN.
DIRECT = DIRECTION OF THE REACTION (FORW. = 1.0 , BACK. = 2.0)
AIJ = FACTOR TO ALLOW USE OF GENERAL SPECIES IN REACTION EQNS.
i.e. (N2 + M = 2N + M), AIJ = 1.0 OR (N2 + O2 = 2NO), AIJ = 0.0
NUIJ = STOICHIOMETRIC COEFFICIENTS OF I-TH REACTANT IN J-TH REACTION
NUIJ = STOICHIOMETRIC COEFFICIENT FOR I-TH PRODUCT IN J-TH REACTION
ALPIK = FACTORS IN EQN. FOR VIBRATIONAL RELAXATION TIME
BETAIK = FACTORS IN EQN. FOR VIBRATIONAL RELAXATION TIME

```

C      SIGIK  = FACTORS IN EQN. FOR VIBRATIONAL RELAXATION TIME
C      NIP    = NUMBER OF VIBRATIONAL LEVELS FOR ANHARMONIC OSCILLATOR , K
C      UP     = CHARACTERISTIC PROBABILITY TEMPERATURE , K
C      WE     = CONST. USED TO COMPUTE ANHARMONIC OSCILLATOR , K
C      WEXE   = CONST. USED TO COMPUTE ANHARMONIC OSCILLATOR , K
C      WEYE   = CONST. USED TO COMPUTE ANHARMONIC OSCILLATOR , K
C      WEZE   = CONST. USED TO COMPUTE ANHARMONIC OSCILLATOR , K
C      XI     = INITIAL COMPUTING INTERVAL , .0001220703125 UNLESS INPUT , cm
C      ELE1   = (2*IMAX + 1) VALUES USED BY INTEGRATION SCHEME
C              NORMALLY 0.1,0.5, OR .05
C      ELE2   = (2*IMAX + 1) VALUES USED BY INTEGRATION SCHEME
C              NORMALLY .05,.1, OR .01 AND .LT. ELE1
C      XPST   = 99 OR LESS XS AT WHICH PHYSICAL SPACE CALCULATIONS ARE
C              DESIRED. THEY MUST BE MULTIPLES OF DELX IN ORDER TO HAVE
C              RS,COST,ZS, AND SINT VALUES AND LAST MUST BE .GT. X AT
C              ZSTERM. THEREFORE, XPST(NXPST) SET = X AT ZSTERM + 100.0
C              IN PROGRAM. XPST(1) MAY NOT BE 0.0 . THEREFORE, SET
C              XPST(1) = DELX IN PROGRAM
C      NXPST  = NUMBER OF X-S AT WHICH PHYSICAL SPACE CALCULATIONS ARE
C              DESIRED
C      CIMAX  = MAX. CJ OR COMPUTING INTERVAL 0.0625 UNLESS INPUT OTHERWISE
C      HCHCKT = CONTROL ON SIZE OF COMPUTING INTERVAL IN CHECK
C              IF(ABS(HPREV - H)/H.GT.HCHECK) REDUCE INTERVAL
C      TCHCKT = CONTROL ON SIZE OF COMPUTING INTERVAL IN CHECK
C              IF(ABS(TPREV - T)/T.GT.TCHECK) REDUCE INTERVAL
C      PHMAX  = CONTROL ON COMPUTING INTERVAL .LE. 65.0
C      IPF    = OUTPUT PRINT FREQUENCY

C      STOPS
C      -----
C      STOP 1      INCORRECT INPUT
C      STOP 2      IN MAIN WHEN NXPST IS .LE. 2
C      STOP 13     IN SHOCKG
C      STOP 30     IN CHECK WHEN COMPUTING INTERVAL .LT. 1.0E-15
C      STOP 66     IN BASIC WHEN NO CONVERGENCE ON E ITERATION (FOFE)
C      STOP 301    IN MAIN FOR ERROR IN XPST ARRAY OR IZTERM .GT. 500
C      STOP 321    IN MAIN WHEN NO CONVERGENCE ON TS ITERATION (FOFTS)
C      STOP 663    IN MAIN WHEN X .NE. VARI(IPS1)
C      STOP 665    IN MAIN AFTER AN INTEGRATION ATTEMPT
C      STOP 670    IN MAIN WHEN A CI NEGATIVE

C      IMPLICIT REAL*8(A-H,D-Z)
C      COMMON /A1/ TE1,TE2,M,IX,IMAX,IPSI,MODEL
C      COMMON /A2/ P(500),DPDX(500),VARI(500)
C              FOLLOWING 7 VARIABLES DIMENSIONED BY IMAX
C      COMMON /A3/ EVIINF(25),THETAI(25),MUI(25),FI(25),DELHI(25),
C      *      CIINF(25),LI(25)
C      COMMON /A4/ R,MUINF,DELTA,LAMSQ,OMEGSQ,OMEGA,MU,T,U,TINF
C              FOLLOWING 2 VARIABLES DIMENSIONED BY (LMAX IN LI,IMAX)
C      COMMON /A5/ EPSIIL(20,25),GIL(20,25),CID(200,50)
C      COMMON /A6/ VAR(52),CUVAR(52),DER(51)
C              FOLLOWING 4 VARIABLES DIMENSIONED BY JMAX
C      COMMON /A7/ MJ(50),EJ(50),AJ(50),BJ(50),DIRECT(50)
C              FOLLOWING 3 VARIABLES DIMENSIONED BY IMAX
C      COMMON /A8/ TVI(25),DGENI(25),BI(25),NI(25)
C              EVIS(IMAX)
C      COMMON /A9/ EVIS(25),XPST(100)
C              FOLLOWING 3 VARIABLES DIMENSIONED BY (IMAX+1,JMAX) OR (IMAX,JMAX)
C      COMMON /A10/ NUIJ(26,50),AIJ(25,50),NUIJ(25,50)
C              FOLLOWING 3 VARIABLES DIMENSIONED BY (IMAX,JMAX)
C      COMMON /A11/ SIGIK(25,25),ALPIK(25,25),BETAIK(25,25)
C      COMMON /A12/ SINTM(1500),COSTM(1500),RSM(1500),RCM(1500),X1(1500)
C      *      ,ZSM(1500)
C      COMMON /A13/ SP,TS,DELX,ZSTERM,IZTERM,NSR,MW
C      COMMON /A14/ EINF,PINF,RHOINF,VINF,E,JMAX,KEYINT,RHO,HSTAG
C      COMMON /A15/ PFTL,KITR1,NIP(25),UP(25)
C      COMMON /A16/ WE(25),WEXE(25),WEYE(25),WEZE(25),IUNEG
C      COMMON /A17/ ELB,SPEC,CJ,TPREV,HPREV,HCHECK,TCHECK
C      COMMON /A18/ ITNEG,IEXP
C      COMMON /A19/ ELE1(51),ELE2(51),NERR
C      COMMON /RAD/ RCON(100,10,40),RRHO(100,40),TEMP(100,40)
C      *      ,TSTAG,Y(100,40),NXCON,NPTS(100)
C      COMMON /RADC/ YBDY(100),B(100,8,40),DPTL(100,8,40)
C      COMMON /RADO/ TEMPO(100,40),ALPN(100,40),ALPO(100,40)
C      COMMON /RADOL/ BETN(100,40),BETO(100,40)
C      COMMON /PLOT/ XSTR1(2000),ZSTR1(2000),RSTR1(2000)
C              DELI(IMAX)

C      DIMENSION DELI(25),PREC(25)
C      DIMENSION RUT(200),PSISTG(200)
C      DIMENSION CIG(25,11),EIG(25,11)

```



```

DO 45 J = 1,JMAX,2
45 READ(7,*) MJ(J),AJ(J),BJ(J),EJ(J),DIRECT(J)
DO 48 J = 2,JMAX,2
48 READ(7,*) MJ(J),DIRECT(J)
IMAX21 = 2*IMAX + 1
DO 51 I = 1,IMAX21
51 READ(7,*) ELE1(I),ELE2(I)
DO 54 I = 1,IMAX
54 READ(7,*) (AIJ(I,J),J = 1,JMAX)
DO 57 J = 1,JMAX
57 READ(7,*) (NUIJ(I,J),I = 1,IMAX+1)
DO 60 J = 1,JMAX
60 READ(7,*) (NUPIJ(I,J),I = 1,IMAX)
DO 63 I = 1,IMAX
63 READ(7,*) (ALPIK(J,I),J = 1,IMAX)
DO 66 I = 1,IMAX
66 READ(7,*) (BETAIK(J,I),J = 1,IMAX)
DO 69 I = 1,IMAX
69 READ(7,*) (SIGIK(J,I),J = 1,IMAX)
DO 72 I = 1,IMAX
LII = LI(I)
72 READ(7,*) (GIL(L,I),L = 1,LII)
DO 75 I = 1,IMAX
LII = LI(I)
75 READ(7,*) (EPSIIL(L,I),L = 1,LII)
IF (IANS.EQ.1) THEN
  NXPST = 4
  XPST(1) = (IS3 - NSR)*DELX
  XPST(2) = (IS4 - NSR)*DELX
  XPST(3) = (IS5 - NSR)*DELX
  XPST(4) = (IS6 - NSR)*DELX
  GO TO 78
ENDIF
DO 76 I = 1,NXPST
NPTS(I) = I
76 XPST(I) = DELX*I
78 DO 80 I = 1,IMAX
80 READ(7,*) NIP(I),UP(I),WE(I),WEXE(I),WEYE(I),WEZE(I)
IMAXP2 = IMAX + 2
C SUM OF CIINF SHOULD BE 1.0
SUM = 0.0
DO 81 I = 1,IMAX
81 SUM = SUM + CIINF(I)
IF (SUM.EQ.1.0) GO TO 85
WRITE(15,82) SUM
82 FORMAT(1X,'SUM OF CIINF(I) SHOULD BE 1: IT IS = ',E11.4,' STOP 1')
STOP
C STOP 1
85 HCHECK = HCHCKT
TCHECK = TCHCKT
C READ IN AND WRITE OUT SPECIES AND REACTIONS
DO 88 I = 1,IMAX
88 READ(7,90) SPECIE(I)
90 FORMAT(A30)
WRITE(8,95)
95 FORMAT(//)
WRITE(8,100)
100 FORMAT(14X,'SPECIES'//)
DO 110 I = 1,IMAX
105 FORMAT(1X,I5,11X,A30)
110 WRITE(8,105) I,SPECIE(I)
WRITE(8,95)
DO 112 I = 1,JMAX
112 READ(7,90) REACT(I)
WRITE(8,115)
115 FORMAT(14X,'REACTIONS'//)
DO 120 I = 1,JMAX
120 WRITE(8,125) I,REACT(I)
125 FORMAT(1X,I5,8X,A30)
WRITE(8,95)
REWIND 7
C STORE PRELIMINARY INFO. FOR DISSPLA
WRITE(69,*) IMAX,JMAX,MODEL,VINF,PINF,TINF
C PRINT INPUT
WRITE(8,135) HCHCKT,TCHCKT,PHMAX
135 FORMAT(1X,'HCHCKT = ',E11.4,4X,'TCHCKT = ',E11.4,4X,'PHMAX = ',
* E11.4//)
WRITE(8,140) XI,CIMAX,NXPST
140 FORMAT(1X,'XI = ',E11.4,8X,'CIMAX = ',E11.4,5X,'NXPST = ',I3//)
DO 141 I = 1,IMAX21

```

```

141 WRITE(8,142) I,ELE1(I),I,ELE2(I)
142 FORMAT(1X,'ELE1(',I2,') = ',E11.4,5X,'ELE2(',I2,') = ',E11.4)
WRITE(8,95)
DO 143 I = 1,NXPST
143 WRITE(8,144) I,XPST(I)
144 FORMAT(1X,'XPST(',I3,') = ',E11.4)
WRITE(8,95)
WRITE(8,145) IMAX,JMAX,M
145 FORMAT(1X,'IMAX = ',I2,11X,'JMAX = ',I2,13X,'M = ',I2/)
WRITE(8,150) DELX,ZSTERM,IPF,R,PINF,TINF,VINF,GAMMA
150 FORMAT(1X,'DELX = ',E11.4,2X,'ZSTERM = ',E11.4,2X,
* 'PRINT FREQ. = ',I3,3X,'R = ',E11.4,/,1X,'PINF = ',E11.4,
* 2X,'TINF = ',E11.4,4X,'VINF = ',E11.4,2X,'GAMMA = ',E9.3/)
WRITE(8,155)
155 FORMAT(16X,'MUI',8X,'THETAI',7X,'DGENI',10X,'FI',9X,'DELHI'/)
DO 157 I = 1,IMAX
157 WRITE(8,160) I,MUI(I),THETAI(I),DGENI(I),FI(I),DELHI(I)
160 FORMAT(1X,I2,8X,5(E11.4,2X))
WRITE(8,95)
WRITE(8,161)
161 FORMAT(16X,'DELI',8X,'CIINF',9X,'EVI',10X,'LI',11X,'BI'/)
DO 163 I = 1,IMAX
163 WRITE(8,164) I,DELI(I),CIINF(I),EVI(I),LI(I),BI(I)
164 FORMAT(1X,I2,8X,3(E11.4,2X),3X,I3,7X,E11.4)
WRITE(8,95)
WRITE(8,165)
165 FORMAT(8X,'NIP',9X,'UP',10X,'WE',10X,'WEZE',9X,'WEYE',8X,'WEZE'/)
DO 167 I = 1,IMAX
166 FORMAT(1X,I2,5X,I3,4X,E10.4,2X,E10.4,3X,E10.4,3X,E10.4,2X,E10.4)
167 WRITE(8,166) I,NIP(I),UP(I),WE(I),WEZE(I),WEYE(I),WEZE(I)
WRITE(8,95)
WRITE(8,170)
170 FORMAT(12X,'MJ',9X,'AJ',14X,'BJ',15X,'EJ',10X,'DIRECTION',/,
* 69X,'F=1,B=2'/)
DO 180 J = 1,JMAX,2
175 FORMAT(1X,I2,2X,I8,3(2X,E15.6),5X,F5.2)
180 WRITE(8,175) J,MJ(J),AJ(J),BJ(J),EJ(J),DIRECT(J)
WRITE(8,95)
WRITE(8,181)
181 FORMAT(12X,'MJ',6X,'DIRECTION',/,21X,'F=1,B=2'/)
DO 183 J = 2,JMAX,2
182 FORMAT(1X,I2,2X,I8,8X,F5.2)
183 WRITE(8,182) J,MJ(J),DIRECT(J)
WRITE(8,95)
DO 187 I = 1,IMAX
LII = LI(I)
DO 186 L = 1,LII
WRITE(8,185) L,I,GIL(L,I),L,I,EPSIIL(L,I)
185 FORMAT(1X,'GIL(',I2,',',I2,') = ',E11.4,4X,
* 'EPSIIL(',I2,',',I2,') = ',E11.4)
186 CONTINUE
187 CONTINUE
WRITE(8,95)
DO 200 I = 1,IMAX
DO 196 J = 1,JMAX
WRITE(8,195) I,J,AIJ(I,J)
195 FORMAT(1X,'AIJ(',I2,',',I2,') = ',E11.4)
196 CONTINUE
200 CONTINUE
WRITE(8,95)
DO 205 I = 1,IMAX+1
DO 202 J = 1,JMAX
WRITE(8,201) I,J,NUIJ(I,J),I,J,NUPIJ(I,J)
201 FORMAT(1X,'NUIJ(',I2,',',I2,') = ',I5,9X,'NUPIJ(',I2,',',
* I2,') = ',I5)
202 CONTINUE
205 CONTINUE
WRITE(8,95)
DO 225 I = 1,IMAX
DO 220 J = 1,IMAX
WRITE(8,210) I,J,SIGIK(I,J),I,J,ALPIK(I,J),I,J,BETAIK(I,J)
210 FORMAT(1X,'SIGIK(',I2,',',I2,') = ',E10.4,1X,
* 'ALPIK(',I2,',',I2,') = ',E10.4,1X,'BETAIK(',I2,',',I2,') = ',
* E10.4)
220 CONTINUE
225 CONTINUE
WRITE(8,95)
WRITE(8,230)
230 FORMAT(7X,25('*'), ' END INPUT ',25('*')///)
WRITE(8,231)

```

```

231  FORMAT(25X,'+++ SHOCK GEOMETRY +++'//)
      NXPSTM = NXPST - 1
C
      CALL SHOCKG (NXPSTM,ITK)
      IZTERM = NO. OF DELTA X INCREMENTS GENERATED IN SHOCKG
C
      WRITE(8,95)
      IF ((NXPST - 1.0).LE.IZTERM) GO TO 237
      WRITE(15,235)
235  FORMAT(1X,'(NXPST-1).GT.IZTERM : IT MUST BE .LE. TO IZTERM',/,
*      1X,'PHYSICAL SPACE CALCULATIONS - SEE DO 700 LOOP : STOP 301'//)
      STOP
237  WRITE(8,95)
      IF (IZTERM.LE.500) GO TO 241
      WRITE(15,240)
240  FORMAT(1X,'IZTERM.GT.500:CHGE DIM OF P,DPDX AND VARI: STOP 301'//)
      STOP
C
      FREESTREAM QUANTITIES
241  SUM = 0.0
      DO 245 I = 1,IMAX
245  SUM = SUM + CIINF(I)/MUI(I)
      MUINF = 1.0/SUM
      RHOINF = MUINF*PINF/(R*TINF)
      AINF = DSQRT(GAMMA*R*TINF/MUINF)
      MINF = VINP/AINF
C
      FOR EACH I , SPECIE
      EINF = 0.0
      DO 275 I = 1,IMAX
      TEM = DEXP(THETA(I)/TINF)
      EVIINF(I) = (R*THETA(I))/(MUI(I)*(TEM - 1.0))*FI(I)
C
      FOR EACH L , REACTION LEVEL
      GSUM = 0.0
      GESUM = 0.0
      LII = LI(I)
      IF (LII.LE.20) GO TO 255
      WRITE(15,250) LII
250  FORMAT(1X,'LII = ',I3,2X,'A LEVEL IN LI ARRAY IS GREATER THAN 20',
*      2X,/,1X,'YOU NEED TO CHANGE DIMENSION OF EPSIIL AND GIL')
255  CONTINUE
      DO 265 L = 1,LII
      TEM1 = DEXP(-EPSIIL(L,I)/TINF)
      GSUM = GSUM + GIL(L,I)*TEM1
      GESUM = GESUM + GIL(L,I)*EPSIIL(L,I)*TEM1
265  CONTINUE
      EEIINF = R/MUI(I)*(GESUM/GSUM)
      EIINF = 1.5*R*TINF/MUI(I) + FI(I)*R*TINF/MUI(I) + EVIINF(I)
*      + EEIINF + DELHI(I)/MUI(I)
      EINF = EINF + EIINF*CIINF(I)
275  CONTINUE
      WRITE(8,280) MUINF,RHOINF,AINF,TINF,PINF,VINF,MINF,EIINF,EINF,
*      EEIINF
280  FORMAT(24X,'+++ FREESTREAM QUANTITIES +++',///,6X,'MUINF = ',E11.4
*      ,3X,'RHOINF = ',E11.4,3X,'AINF = ',E11.4,///,6X,'TINF = ',E11.4,3X
*      ,',PINF = ',E11.4,3X,'VINF = ',E11.4,///,1X,'MINF = ',E10.4,2X,
*      ',EEIINF = ',E10.4,2X,'EINF = ',E10.4,2X,'EEIINF = ',E10.4,2X,
      DO 285 I = 1,IMAX
285  WRITE(8,286) I,EVIINF(I)
286  FORMAT(1X,'EVIINF(',I2,') = ',E11.4)
      WRITE(8,95)
C
      BEGIN QUANTITIES BEHIND SHOCK FOR THE RANGE OF X
      WRITE(8,290)
290  FORMAT(22X,'+++ QUANTITIES BEHIND SHOCK +++',///,5X,'TS',9X,'ES',
*      9X,'PS',7X,'RHOS',8X,'US',8X,'PSIS',8X,'HS'//)
      IF (NXPST.GE.3) GO TO 293
      WRITE(15,291)
      STOP
291  FORMAT(1X,'NXPST = 2 - SURELY SOME PHYSICAL SPACE VALUES',/,1X,
*      'ARE DESIRED -- MAKE IT AT LEAST 3 : STOP 2')
C
293  DO 385 IX = 1,IZTERM
      TEM = SINTM(IX)**2
      LAMBDA = RHOINF*VINP*SINTM(IX)
      OMEGA = PINF + RHOINF*VINP**2*TEM
      DELTA = EINF + PINF/RHOINF + (VINP**2*TEM)/2.0
      LAMSQ = LAMBDA**2
      OMEGSQ = OMEGA**2
      MINFSQ = MINF**2
      TEM = TEM*MINFSQ
      TSG = (TINF*(2.0*GAMMA*TEM - (GAMMA - 1.0))*((GAMMA - 1.0)
*      *TEM + 2.0))/((GAMMA + 1.0)**2*TEM)
      TOL1 = .001

```

```

TOL2 = .0001
MAXI = 50

C
IF (IQ3.EQ.1) CALL FOFTS1 (MAXI,TOL1,TOL2,ICODE,TSG)
IF (IQ3.GT.1) CALL FOFT23 (MAXI,TOL1,TOL2,ICODE,TSG,IQ1,IQ2,IQ3)

C
IF (ICODE.EQ.1) WRITE(15,330) TSG
330 FORMAT(1X,'**** MAXIMUM ITERATION EXCEEDED IN FOFTS ****',//,
* 1X,'LAST ITERATED VALUE OF TSG WAS ',E15.6,' : STOP 321'//)
IF (ICODE.EQ.2) WRITE(15,335)
335 FORMAT(1X,'***** DERIVATIVE = 0.0 IN FOFTS ***** : STOP 321'//)
IF (ICODE.EQ.1.OR.ICODE.EQ.2) STOP
STOP 321

C
IF (IX.EQ.1) TSTAG = TSG
TS = TSG
ES = 0.0
DO 360 I = 1,IMAX
IF (M.EQ.1) EVIS(I) = EVIINF(I)
IF (M.EQ.1) GO TO 345
TEM = DEXP(THETAI(I)/TS)
EVIS(I) = (R*THETAI(I))/(MUI(I)*(TEM - 1.0))*FI(I)
345 SUMG = 0.0
SUMGE = 0.0
LII = LI(I)
DO 350 L = 1,LII
TEM1 = DEXP(-EPSIIL(L,I)/TS)
SUMG = SUMG + TEM1*GIL(L,I)
350 SUMGE = SUMGE + TEM1*GIL(L,I)*EPSIIL(L,I)
EEIS = (R/MUI(I))*(SUMGE/SUMG)
EIS = (1.5*R*TS)/MUI(I) + (FI(I)*R*TS)/MUI(I) + EVIS(I)
* + EEIS + DELHI(I)/MUI(I)
IF (IQ3.EQ.1) THEN
ES = ES + EIS*CIINF(I)
ELSE
ES = ES + EIS*CID(IX,I)
ENDIF
360 CONTINUE
PS = DSQRT(OMEGSQ - 2.0*LAMSQ*(DELTA - ES))
RHOS = (MUIINF*PS)/(R*TS)
US = VINFCOSTM(IX)
PSIS = (RHOINF*VINFCRSM(IX)**2)/2.0
HS = PS/RHOS + ES
WRITE(10,*) IX,TS,ES,PS,RHOS,US,PSIS,HS
WRITE(8,365) TS,ES,PS,RHOS,US,PSIS,HS
365 FORMAT(1X,7(E9.4,2X))
WRITE(9,366) IX
366 FORMAT(1X,'FOR IZTERM = ',I3//)
DO 370 I = 1,IMAX
WRITE(9,367) I,EVIS(I)
367 FORMAT(1X,'EVIS(',I2,') = ',E11.4)
370 CONTINUE
WRITE(11,*) (EVIS(I),I = 1,IMAX)
WRITE(9,620)
385 CONTINUE

C
REWIND 9
REWIND 10
REWIND 11

C
COMPUTE PRESSURE DISTRIBUTION FOR EACH PSI
DO 390 IPSI = 1,IZTERM
DO 388 IX = 1,IZTERM
IF (IX.EQ.IPSI) XX = X1(IX)
READ(10,*) IXT,TS,ES,PS,RHOS,US,PSIS,HS
IF (IXT.LT.IPSI) GO TO 388
IF (IXT.EQ.IPSI) PSISHK = PSIS
IF (IPSI.EQ.IX) THEN
P(IX) = PS
GO TO 388
ENDIF
P(IX) = PS + US/(RCM(IX)*RSM(IX))*(PSISHK - PSIS)
388 CONTINUE
REWIND 10
WRITE(12,*) XX,(P(IX),IX = 1,IZTERM),IPSI,IZTERM
390 CONTINUE
REWIND 12

C
COMPUTE DPDX AND VARI
DO 404 IPSI = 1,IZTERM
READ(12,*) X,(P(I),I=1,IZTERM),IPI,IZTERM
ITHIS = 0
DO 401 IX = IPSI,IZTERM

```

```

      IF (IX.EQ.IPSI) XT = X
      IF (IX.GT.IPSI.AND.IPSI.LT.(NSR+2)) .XT = XT + DELX/NSR
      IF (IX.GT.IPSI.AND.IPSI.GT.(NSR+1)) XT = XT + DELX
      ITHIS = ITHIS + 1
      VARI(ITHIS) = XT
401 CONTINUE
      DO 403 IX = IPSI,IZTERM
C
        IF (IX.NE.IPSI) GO TO 402
        DPDX(IX) = (- 3.*P(IX) + 4.*P(IX+1) - P(IX+2))/(2.0*DELX)
        GO TO 403
402 DPDX(IX) = (- P(IX-1) + P(IX+1))/(2.0*DELX)
        IF (IX.NE.IZTERM) GO TO 403
        DPDX(IX) = (P(IX-2) - 4.*P(IX-1) + 3.*P(IX))/(2.0*DELX)
403 CONTINUE
        WRITE(13,*) (DPDX(I),I=1,IZTERM),IPSI,IZTERM
404 CONTINUE
        REWIND 13
        REWIND 12
C
        DO 405 IX = 1,IZTERM
        IF (IX.EQ.1) VARI(1) = 0.0
        IF (IX.EQ.1) GO TO 405
        IF (IX.LT.(NSR+2)) VARI(IX) = VARI(IX-1) + DELX/NSR
        IF (IX.GT.(NSR+1)) VARI(IX) = VARI(IX-1) + DELX
405 CONTINUE
        WRITE(8,95)
        WRITE(8,407) ITK,NXPST
407 FORMAT(1X,'XXXXX' ITK = ',I3,5X,'NXPST = ',I3,' XXXXX'//)
        IF (ITK.GE.(2*NXPST/4)) GO TO 409
        WRITE(15,408)
408 FORMAT(1X,'NO. OF X-S IN XPST ARRAY,ITK, IS .LT. .5*NXPST',/
        * ,1X,'REEXAMINE DELX,NXPST AND XPST ARRAY : STOP 301'/)
        STOP
C
C                                     STOP 301
C
END COMPUTATION OF QUANTITIES BEHIND SHOCK ; INTEGRATE ALONG EACH PSI
C
VAR(1) = XVAR
VAR(2) = H,ENTHALPY
VAR(2) MUST BE H, WHICH MAY BE + OR -
C
CONCENTRATION OF SPECIE
VAR(3) = CI(1)
VAR(2+IMAX) = CI(IMAX)
EQUILIBRIUM VIBRATIONAL ENERGY
VAR(2+IMAX+1) = EVI(1)
VAR(2+IMAX+IMAX) = EVI(IMAX)
C
409 WRITE(8,411)
411 FORMAT(1X,'NO. OF VIB. LEVELS TO DISSOCIATION',/)
DO 410 I = 1,IMAX
C
C                                     TRUNCATE
YIY = FI(I)*(DELI(I)/THETAI(I) + 1.0)
NI(I) = AINT(YIY)
410 WRITE(8,412) I,NI(I)
412 FORMAT(11X,'NI(',I2,',') = ',I3)
WRITE(8,95)
MU = 0.0
N = 1 + 2*IMAX
IF (M.EQ.0) N = N - IMAX
C
C                                     INITIALIZE
WRITE(8,415) IZTERM
415 FORMAT(1X,'IZTERM = ',I4///)
WRITE(8,417)
417 FORMAT(1X,'COUPLED VIBRATION-DISSOCIATION MODELS',/,12X,
* 'TYPE',11X,'NO.',/,9X,'VIB. EQUIL.',8X,'O',/,12X,'CVD',13X,
* '1',/,12X,'CVDV',12X,'2',/,6X,'CVDV-Preferential',5X,'3',/,
* 12X,'PARK',12X,'4'///)
WRITE(8,418)
418 FORMAT(7X,'SHOCK JUMP CONDITION MODELS',/,12X,
* 'TYPE',11X,'NO.',/,6X,'CHEMISTRY FROZEN',6X,'1',/,5X,
* 'N2 FROZEN, O2 DISS.',4X,'2',/,7X,'N2 AND O2 DISS.',6X,'3'///)
IF (MW.EQ.1.AND.IMAX.EQ.7) THEN
  WRITE(8,419)
419 FORMAT(3X,'MILLIKAN AND WHITE DATA FOR N2 WAS SELECTED'//)
ENDIF
IF (MW.EQ.1.AND.IMAX.EQ.10) THEN
  WRITE(8,426)
426 FORMAT(3X,'MILLIKAN AND WHITE DATA FOR N2 AND N2+ WAS SELECTED'//)
ENDIF

```

```

KSTAG = 0
ISTAG = 0
XPST(1) = DELX
XPST(NXPST) = VARI(IZTERM) + 100.0
SUMCI = 0.0
WRITE(8,95)
LPS1 = 1
IPSI = 0

C                               BEGIN EACH STREAMLINE COMPUTATION HERE
420 IPSI = IPSI + 1
    IF (IPSI.EQ.IZTERM) GO TO 735
425 KIPF = 0
C                               ISTAG = 2 FOR STREAMLINE DELX
    IF (ISTAG.EQ.1) ISTAG = 2
    READ(10,*) IX,TS,ES,PS,RHOS,US,PSIS,HS
    READ(11,*) (EVIS(I),I = 1,IMAX)
    IF (IPSI.EQ.1) HSTAG = HS
    CJ = XI
    SPEC = 0.0
C                               EVALUATE DERIVATIVES WHEN SPEC = 0.0
    II = 0
C                               USE SHOCK VALUES FOR INITIAL COMPUTATION ON EACH STREAMLINE
    KEYINT = 0
    IX = 0
    T = TS
    IF (M.EQ.0) GO TO 450
    DO 445 I = 1,IMAX
445 TVI(I) = TINF
    TE1=TINF
    GO TO 460
450 DO 455 I = 1,IMAX
455 TVI(I) = TS
    TE1=TINF
460 RHO = RHOS
    IF (IQ3.GT.1) THEN
        SUM = 0.0
        DO 462 I = 1,IMAX
462 SUM = SUM + CID(IPSI,I)/MUI(I)
        MU = 1.0/SUM
    ENDIF
    IF (IQ3.EQ.1) MU = MUINF
    U = US
    DO 465 I = 1,IMAX
    MM = 2 + I
    K = IMAX + MM
    VAR(MM) = 0.0
    IF (IQ3.EQ.1) VAR(MM) = CIINF(I)
    IF (IQ3.GT.1) VAR(MM) = CID(IPSI,I)
    VAR(K) = 0.0
    IF (M.EQ.0) VAR(K) = EVIS(I)
    IF (M.EQ.1) VAR(K) = EVIINF(I)
    EVI(I) = VAR(K)
465 CONTINUE
C                               COMPUTE S EXPONENT FOR PARK MODEL
    IF (MODEL.EQ.4) THEN
        SP = 3.5*DEXP(-5000.0/TS)
    ENDIF
    E = ES
    H = HS
    IF ((IPSI.EQ.(NSR+1)).AND.(ISTAG.EQ.0)) GO TO 635
    READ(12,*) X,(P(L),L=1,IZTERM),IPSI,IZTERM
    READ(13,*) (DPDX(L),L=1,IZTERM),IPSI,IZTERM
    VAR(1) = X
C                               OMIT PSI = 0.0 STREAMLINE FOR NOW - PICK IT UP LATER
    IF (X.NE.0.0) GO TO 490
    IG = 0
    WRITE(8,725)
    GO TO 730
490 IF (IANS.EQ.1) THEN
    IF ((IPSI.EQ.IS1).OR.(IPSI.EQ.IS2).OR.(IPSI.EQ.IS3)) GO TO 492
    IF ((IPSI.EQ.IS4).OR.(IPSI.EQ.IS5).OR.(IPSI.EQ.IS6)) GO TO 492
    GO TO 730
    ENDIF
492 IF (DABS(X - VARI(IPSI)).LE.1.0E-06) GO TO 500
    WRITE(15,495) IPSI,X,VARI(IPSI)
495 FORMAT(1X,'IPSI = ',I3,2X,'X = ',E10.5,2X,'VARI(IPSI) = ',E10.5,
*//,1X,'X AND VARI(IPSI) SHOULD BE EQUAL : EXAMINE GENERATION OF X
* AND VARI IN MAIN : STOP 663'/)
    STOP
C

```

STOP 663

```

C                               INITIALIZE
500 IF (IANS.EQ.1) THEN
    IF ((IPSI.EQ.IS1).OR.(IPSI.EQ.IS2).OR.(IPSI.EQ.IS3)) GO TO 501
    IF ((IPSI.EQ.IS4).OR.(IPSI.EQ.IS5).OR.(IPSI.EQ.IS6)) GO TO 501
    GO TO 730
ENDIF
501 ELB = 0.0
    SPEC = 0.0
    IF (VAR(1).GT.XPST(LPS1)) LPS1 = LPS1 + 1
    LPS2 = LPS1
505 IX = IX + 1

C                               CALITH - MODIFIED RUNGE-KUTTA
    CALL CALITH (N,CIMAX,PHMAX)

C
    IF (NERR.EQ.0) GO TO 575
    IF (NERR.EQ.1) WRITE(15,565) CJ,N
565 FORMAT(1X,'BAD INPUT IN CALITH',/,1X,'CJ = ',E10.5,2X,'N = ',I4
    *      ,/,1X,'STOP 665'/)
    IF (NERR.EQ.2) WRITE(15,570)
570 FORMAT(1X,'EXAMINE ELE1 AND ELE2 IN CALITH : STOP 665'/)
    IF ((NERR.EQ.1).OR.(NERR.EQ.2)) STOP

C                               STOP 665
575 IF (IX.EQ.1) THEN
    IF (IQ3.NE.1) THEN
        IF (IQ2.EQ.1) TE = TINF
        IF (IQ2.EQ.2) TE = TS
        TE1 = TE
        TE2 = TE
    ELSE
        TE = TINF
        TE1 = TE
        TE2 = TE
    ENDIF
ENDIF

C                               ELECTRON TEMPERATURE MODEL
    IF (IX.EQ.1) GO TO 576
    CORR = 4.23E-06*(T**(-2.88))
    CN = RHO*6.02252E+23
    CNN = CN*VAR(5)/MUI(3)
    CNO = CN*VAR(6)/MUI(4)
    TE = DLOG(1. + 10.1/((CNO+CNN)*CORR))
    TE = T/(1. + TE*T/85000.)
    TE1 = TE
    ITRTE=0
    FNA=CN*VAR(5)/MUI(3)
    FNI=CN*VAR(11)/MUI(9)
    FNE=CN*VAR(9)/MUI(7)
    FNM=CN*VAR(3)/MUI(1)
    FOA=CN*VAR(6)/MUI(4)
    FOI=CN*VAR(12)/MUI(10)
    FOM=CN*VAR(4)/MUI(2)
    EI=2.3322D-11
    EO=2.1735D-11
    FAV=6.023D23
    SA=1.52D-15
    F1=0
    F2=0
    FO=0
574 ITRTE=ITRTE+1
    IF(ITRTE.GT.1)GO TO 571
    KK=1
569 KK=KK+1
    IF(KK.GT.50)TE1=2000.D00
    IF(KK.GT.50)GO TO 576
    TE=1000*KK
571 SM=0.5355D-19*TE+.696D-15
    IF(ITRTE.GT.50)GO TO 573
    SI=1.53614D08*(TE**3)/FNE
    SI=DSQRT(SI)
    SI=((4.38384D-06)/(TE**2))*DLOG(SI)
    FKF=4.16D13*(TE** ( 0.50D0))*DEXP(-1.20D05/TE)
    FK1=5.49D13*(TE** ( 0.50D0))*DEXP(-1.045D05/TE)
    FKB=2.3D21*(TE**(-1.0D00))*DEXP(4.9D4/TE)
    FK5=1.53D22*(TE**(-1.0D00))*DEXP(5.3D4/TE)
    WEA=FKF*FNA-FKB*(FNI/FAV)*FNE
    W2E=FK1*FOA-FK5*(FOI/FAV)*FNE
    SX=(FNA+FOA)*SA+(FNI+FOI)*SI+(FNM+FOM)/(2.0D0)*SM
    W2=DER(8)*U*FAV/VAR(9)
    W1=WEA+W2E
    W1=W2

```



```

FE=T-1.23357D-10/SX/DSQRT(TE)*(WEA*EI+W2E*EO+W1*3.45D-16*TE)-TE
IF(ITRTE.GT.1)GO TO 568
IF((FE/DABS(FE)*FO).LT.O.DOO)GO TO 567
FO=FE
TO=TE
GO TO 569
567 F1=FE
T1=TE
566 TE=(TO*F1-T1*FO)/(F1-FO)
IF(DABS(TE-T1).LT.1.DOO)GO TO 573
IF(DABS(TE-TO).LT.1.DOO)GO TO 573
GO TO 574
568 IF((FE/DABS(FE)*FO/DABS(FO)).LT.O.OOOO)GO TO 564
TO=TE
FO=FE
GO TO 566
564 F1=FE
T1=TE
GO TO 566
573 IF(TE.GT.T)TE=T
C 573 CONTINUE
TE1=TE
IF(ITRTE.GT.50)WRITE(6,572)
572 FORMAT(' ','TE DID NOT CONVERGE')
CN2 = RHO*6.02252E+23*VAR(3)/MUI(1)
A = DLOG(1.01D+01) - DLOG(CN2) - DLOG(6.4D+00)
* - 59.*DLOG(1.0D+01) + 19.*DLOG(TVI(1))
A = DEXP(A)
TE = DLOG(1. + A)
TE = TVI(1)/(1. + TE*TVI(1)/85000.)
TE2 = TE
C
576 IF (IX.EQ.1) GO TO 580
IF ((IPSI.EQ.(NSR+1)).AND.(IX.EQ.(NSR+1))) GO TO 580
IF (VAR(1).GE.VARI(IZTERM)) GO TO 580
IF ((IPSI.LT.(NSR+1)).AND.(VAR(1).GE.DELX)) GO TO 580
KIPF = KIPF + 1
IF (KIPF.NE.IPF) GO TO 625
KIPF = 0
580 SUMCI = 0.0
DO 590 ISUM = 3,IMAXP2
IF (VAR(ISUM).LT.O.O) NEG = -1
IF (VAR(ISUM).LT.O.O) WRITE(15,585)
585 FORMAT(5X,'----- NEGATIVE CI -----'/)
590 SUMCI = SUMCI + VAR(ISUM)
DO 595 ICM = 1,IMAX
595 CM(ICM) = VAR(ICM+2)*MU/MUI(ICM)
C
WRITE(8,600) IPSI,IX,IQ3,MODEL,XVAR,H,MU,PFTL,RHO,U,T,E,SPEC,SUMCI
600 FORMAT(1X,'PSI = ',I4,3X,'IX = ',I4,8X,'SHOCK J COND. = ',I2,14X,
*'CVD MODEL = ',I2,/,1X,'X = ',D9.4,2X,'H = ',D9.4,2X,'MU = ',
*E9.4,2X,'P = ',E9.4,3X,'RHO = ',E9.4,/,1X,'U = ',E9.4,2X,'T = ',
*E9.4,2X,'E = ',E9.4,2X,'CIT = ',E9.4,2X,'SUMCI = ',E9.4/)
IF (IQ7.EQ.IPSI) THEN
WRITE(15,601) XVAR,T,CM(2),CM(4),SPEC
601 FORMAT(1X,E9.4,2X,E9.4,2X,E9.4,2X,E9.4,2X,E9.4,2X,E9.4)
ENDIF
DO 603 I = 3,IMAXP2
J = I - 2
603 WRITE(8,605) J,VAR(I),J,CM(J),J,VAR(IMAX+I)
605 FORMAT(6X,'CI('',I2,'') = ',D11.4,4X,'CM('',I2,'') = ',E11.4,4X,
* 'EVI('',I2,'') = ',D11.4)
WRITE(8,620)
620 FORMAT(/)
DO 621 I = 1,IMAX
IF (EVI(I).EQ.O.O) TVI(I) = 0.0
IF (I.EQ.IMAX) THEN
WRITE(8,623) I,TVI(I),TE1,ITRTE,W1
GO TO 621
ENDIF
WRITE(8,622) I,TVI(I)
621 CONTINUE
622 FORMAT(5X,'TVI('',I2,'') = ',E11.4)
623 FORMAT(5X,'TVI('',I2,'') = ',E11.4,1X,'TE1= ',E11.4,1X,
* 'ITRTE=',I2,1X,'W1=',E10.3)
WRITE(8,620)
IF (NEG.EQ.-1) WRITE(15,630)
C
STORE INFO FOR DISSPLA
625 IF (IPSI.EQ.IS7) THEN
DO 626 ICM = 1,IMAX

```

```

626 CM(ICM) = VAR(ICM+2)*MU/MUI(ICM)
    WRITE(70,*) (CM(I),I=1,IMAX)
    WRITE(71,*) T,TVI(1),XVAR,TE1
    ENDIF
C
630 FORMAT(1X,' A NEGATIVE CONCENTRATION , CI IN MAIN : STOP 670'//)
    IF (NEG.EQ.-1) STOP
C
C WHEN IPSI = (NSR+1), DELX AND STAGNATION STREAMLINES WILL BE COMPUTED
C IUNEG = 1 IF U**2 NEG. IN BASIC
C
    IF (IUNEG.EQ.1) GO TO 735
    IF ((IPSI.LT.(NSR+1)).AND.(VAR(1).LT.XPST(1))) GO TO 505
    IF ((ISTAG.EQ.1).OR.(ISTAG.EQ.2)) GO TO 690
C
C     ISTAG = 0 UNTIL AFTER EXTRAPOLATION FOR PSI = 0.0 STREAMLINE
C     ISTAG = 1 FOR PSI = 0.0 STREAMLINE
C     ISTAG = 2 FOR DELX STREAMLINE AND THEREAFTER
635 IG = IG+1
    IGC = 0
    IGE = IMAX + 2
    PSIG(IG) = PSIS
    DO 640 IG1 = 3,IMAXP2
        IGC = IGC + 1
        CIG(IGC,IG) = VAR(IG1)
        IGE = IGE + 1
        EIG(IGC,IG) = VAR(IGE)
640 CONTINUE
    EG(IG) = E
    TG(IG) = T
    HG(IG) = H
    IF (IPSI.LT.(NSR+1)) WRITE(8,725)
    IF (IPSI.LT.(NSR+1)) GO TO 730
C
C     IPSI = (NSR+1) IS THE STAGNATION STREAMLINE ;
C     AT DELX ,EXTRAPOLATE FOR PSI = 0.0 VALUES
C
    PSI = 0.0
    MG = 1
    CALL FTLUP (PSI,E,MG,IG,PSIG,EG)
    CALL FTLUP (PSI,T,MG,IG,PSIG,TG)
    CALL FTLUP (PSI,H,MG,IG,PSIG,HG)
    DO 650 IG2 = 1,IMAX
        DO 645 IG1 = 1,NSR1
            CIT(IG1) = CIG(IG2,IG1)
645 EIT(IG1) = EIG(IG2,IG1)
            CALL FTLUP (PSI,CI(IG2),MG,IG,PSIG,CIT)
            IF (CI(IG2).LT.0.0) CI(IG2) = 1.0E-08
            CALL FTLUP (PSI,EVI(IG2),MG,IG,PSIG,EIT)
650 VAR(2+IMAX*IG2) = EVI(IG2)
C
C     INITIALIZE FOR PSI = 0.0 STREAMLINE
C
    CJ = XI
    SPEC = 0.0
    II = 0
    KEYINT = 0
    IX = 0
    SUM = 0.0
    DO 660 I = 1,IMAX
        SUM = SUM + CI(I)/MUI(I)
        IF (EVI(I).EQ.0.0) TVI(I) = 0.0
        IF (EVI(I).EQ.0.0) GO TO 660
        XL = ((DGNI(I)*R*THETAI(I))/(MUI(I)*EVI(I)) + 1.0)
        ALN = DLOG(XL)
        TVI(I) = THETAI(I)/ALN
660 CONTINUE
    MU = 1.0/SUM
    U = DSQRT(2.0*(HSTAG - H))
    VAR(1) = DELX
C
C     COMPUTE P AND DPDX AT IPSI = (NSR+1) (AT THE BODY)
C
    IF (IANS.EQ.0) WRITE(8,665)
    IF ((IPSI.EQ.IS1).OR.(IPSI.EQ.IS2).OR.(IPSI.EQ.IS3)) WRITE(8,665)
    IF ((IPSI.EQ.IS4).OR.(IPSI.EQ.IS5).OR.(IPSI.EQ.IS6)) WRITE(8,665)
665 FORMAT(1X,'-- PSI = 0.000 STREAMLINE --'//)
    P(NSR+1) = PS + US/(RCM(NSR+1)*RSM(NSR+1))*(-PSIS)
    RHO = P(NSR+1)/(H - E)
    NSR2 = NSR + 2
    DO 670 I = NSR2,IZTERM
        READ(10,*) IXT,TS,ES,PS,RHOS,US,PSIS,HS
670 P(I) = PS + US/(RCM(I)*RSM(I))*(-PSIS)
        REWIND 10
        REWIND 11
        DPDX(NSR+1) = (-3.*P(NSR1)+4.*P(NSR2)-P(NSR3))/(2.*DELX)
C
C     EVALUATE DPDX FROM (NSR+2) TO IZTERM
C
    DO 680 I = NSR2,IZTERM

```

```

      IF (I.LT.IZTERM) THEN
        DPDX(I) = (- P(I-1) + P(I+1))/(2.*DELX)
      ELSE
        DPDX(I) = (P(I-2) - 4.*P(I-1) + 3.*P(I))/(2.*DELX)
      ENDIF
680  CONTINUE
      DO 685 I = 1,NSR
        READ(10,*) IX,TS,ES,PS,RHOS,US,PSIS,HS
685  READ(11,*) (EVIS(J),J = 1,IMAX)
        ISTAG = 1
        PSIS = 0.0
        GO TO 500
      C      AT EACH X WHERE PHYSICAL SPACE CALCULATIONS ARE DESIRED SAVE
      C      1./(RHO*U) ON EACH STREAMLINE
690  IF (XPST(LPS2).EQ.XPST(1)) ICNT = IPSI
      JJJ = IPSI - ICNT + 1
      IF (VAR(1).LT.XPST(LPS2)) GO TO 720
      IF (VAR(1).GT.XPST(LPS2)) GO TO 705
      C      VAR(1) = XPST(LPS2)

      RHOURT = 1.0/(RHO*U)
      TEMP(LPS2,JJJ) = TVI(1)
      TEMPO(LPS2,JJJ) = T
      DO 691 III = 1,IMAX
691  RCON(LPS2,III,JJJ) = VAR(III+2)
      RRHO(LPS2,JJJ) = RHO
      WRITE(14,*) IPSI,RHOURT,VAR(1),PSIS
      GO TO 715
705  YPSIS = PREPSI + (XPST(LPS2) - PREX)*((PSIS - PREPSI)/
      * (VAR(1) - PREX))
      RHOURT = PRERU + (XPST(LPS2) - PREX)*((RHO*U - PRERU)/
      * (VAR(1) - PREX))
      TEMP(LPS2,JJJ) = PRETMP + (XPST(LPS2)-PREX)*((TVI(1)-PRETMP)/
      * (VAR(1) - PREX))
      TEMPO(LPS2,JJJ) = PRET + (XPST(LPS2)-PRET)*((T-PRET)/
      * (VAR(1)-PRET))
      DO 706 III = 1,IMAX
706  RCON(LPS2,III,JJJ) = PREC(III)+(XPST(LPS2)-PREX)*((VAR(III+2)
      * -PREC(III))/(VAR(1)-PREX))
      RRHO(LPS2,JJJ) = PRERHO + (XPST(LPS2)-PREX)*((RHO-PRERHO)/
      * (VAR(1) - PREX))
      RHOURT = 1.0/RHOURT
      WRITE(14,*) IPSI,RHOURT,XPST(LPS2),YPSIS
715  LPS2 = LPS2 + 1
      KSTAG = KSTAG + 1
720  PREX = VAR(1)
      PRERU = RHO*U
      PREPSI = PSIS
      PRETMP = TVI(1)
      PRET = T
      DO 721 III = 1,IMAX
721  PREC(III) = VAR(III+2)
      PRERHO = RHO
      IF (VAR(1).LT.VARI(IZTERM)) GO TO 505
      C      END OF STREAMLINE
      WRITE(8,725)
725  FORMAT(1X,'XXXXXXXXXXXXXXXXX'/)
      IF ((IPSI.EQ.(NSR+1)).AND.(ISTAG.EQ.1)) GO TO 425
730  GO TO 420
      C      COMPUTE PHYSICAL SPACE VALUES
735  REWIND 10
      REWIND 11
      REWIND 12
      REWIND 13
      WRITE(8,740) KSTAG
740  FORMAT(1X,'BEGIN PHYSICAL SPACE CALCULATIONS   KSTAG = ',I5/)
      C
      LPS3 = 0
      IXSTR7 = 0
      DO 840 LPS = 1,NXPST
      REWIND 14
      IK = 0
745  LPS3 = LPS3 + 1
      IF (LPS3.GT.IZTERM) GO TO 840
      IF (DABS(X1(LPS3) - XPST(LPS)).GT.1.0E-08) GO TO 745
      DO 780 IPS = 1,KSTAG
      READ(14,*) IPSI,RHOURT,VAR(1),PSIS
      VARMX = VAR(1) - XPST(LPS)
      IF (DABS(VARMX).GT.1.0E-08) GO TO 780
      IK = IK + 1
      IBOB = IK - 1

```

```

      IF (IK.GT.200) WRITE(15,770) IK
770  FORMAT(1X,'IK.GT.200.CHANGE DIMENSION OF RUT AND PSISTG,IK = ',I3)
      RUT(IK) = RHOURT
      PSISTG(IK) = PSIS
780  CONTINUE
C
      FIND SMALLEST DELTA PSI
      SMALL = 200.0
      DO 785 I = 2,IK
      IF ((PSISTG(I) - PSISTG(I-1)).LT.SMALL) SMALL = PSISTG(I)
      *   - PSISTG(I-1)
785  CONTINUE
      XDEL = PSISTG(IK)/SMALL
C
      TRUNCATE
      L = INT(XDEL + 1.0)
C
      REMAINDERING   M = 0 FOR EVEN, M = 1 FOR ODD
      M = MOD(L,2)
C
      MAKE L EVEN
      IF (M.NE.0) L = L + 1
C
      FIND INTEGRAL 1./(RHO*U) DELTA PSI FROM BODY TO SHOCK
      USING SIMPSONS RULE ; L INCREMENTS , L+1 POINTS
      FL = L
      DPSIS = PSISTG(IK)/FL
      PSII = 0.0
      RB = RUT(1) + RUT(IK)
      DO 800 I = 2,L,2
      PSII = PSII + DPSIS
      CALL FTLUP (PSII,RU1,1,IK,PSISTG,RUT)
      IF (I.EQ.L) RB = RB + 4.0*RU1
      IF (I.EQ.L) GO TO 800
      PSII = PSII + DPSIS
      CALL FTLUP (PSII,RU2,1,IK,PSISTG,RUT)
      RB = RB + 4.0*RU1 + 2.0*RU2
800  CONTINUE
      RB = RB*DPSIS/3.0
      ARR = DSQRT(RSM(LPS3)**2 - 2.0*COSTM(LPS3)*RB)
      YI = (RSM(LPS3) - ARR)/COSTM(LPS3)
      YBDY(IBOB)=YI
      ZE = ZSM(LPS3) + YI*SINTM(LPS3)
      WRITE(8,815) PSISTG(1),X1(LPS3),ARR,YI,ZE
815  FORMAT(1X,'PSI = ',E12.5,/,1X,'X = ',E13.6,3X,'R = ',E13.6
      *   ,3X,'Y = ',E13.6,3X,'Z = ',E13.6/)
C
      STORE BODY PTS. FOR DISSPLA
      WRITE(72,*) ZE,ARR
C
      DO 830 I = 2,IK
      DPSI = PSISTG(I) - PSISTG(I-1)
C
      TRAPEZOIDAL RULE
      TR = (DPSI/2.0)*(RUT(I) + RUT(I-1))
      RB = RB - TR
      ARR = DSQRT(RSM(LPS3)**2 - 2.0*COSTM(LPS3)*RB)
      YI = (RSM(LPS3) - ARR)/COSTM(LPS3)
      Y(IBOB,I-1) = YI
      ZE = ZSM(LPS3) + YI*SINTM(LPS3)
      WRITE(8,815) PSISTG(I),X1(LPS3),ARR,YI,ZE
C
      STORE STRLNE AND SHOCK COORD. FOR DISSPLA
      IF (I.EQ.IK) THEN
      IF ((I+NSRM1).EQ.IS7) THEN
      IXSTR7 = IXSTR7 + 1
      WRITE(73,*) ZSM(IS7),RSM(IS7),X1(IS7)
      XSTR1(IXSTR7) = X1(IS7)
      ZSTR1(IXSTR7) = ZSM(IS7)
      RSTR1(IXSTR7) = RSM(IS7)
      ENDIF
      WRITE(74,*) ZE,ARR
      GO TO 830
      ENDIF
      IF ((I+NSRM1).EQ.IS7) THEN
      IXSTR7 = IXSTR7 + 1
      WRITE(73,*) ZE,ARR,X1(LPS3)
      XSTR1(IXSTR7) = X1(LPS3)
      ZSTR1(IXSTR7) = ZE
      RSTR1(IXSTR7) = ARR
      ENDIF
C
830  CONTINUE
      WRITE(8,835)
835  FORMAT(//)
840  CONTINUE
C
      INTERPOLATE X COORD. TO DETERMINE (Z,R) AND XS
      STRL1 = 0.0

```

```

PRER11 = 0.0
PREZ11 = 0.0
PRESTL = 0.0
REWIND 71
841 READ(71,*,END=842) T11,TVN21,XS1,TE1
CALL FTLUP (XS1,R11,2,IXSTR7,XSTR1,RSTR1)
CALL FTLUP (XS1,Z11,2,IXSTR7,XSTR1,ZSTR1)
SEG = DSQRT((PRER11 - R11)**2 + (PREZ11 - Z11)**2)
IF (XS1.EQ.X1(IS7)) SEG = 0.0
STRL1 = PRESTL + SEG
WRITE(75,*) XS1,Z11,R11,STRL1
PRESTL = STRL1
PRER11 = R11
PREZ11 = Z11
GO TO 841

```

RADIATION MODELS

```

C
842 IF (IQ4.EQ.0) GO TO 845
IF (IQ5.EQ.0) WRITE(8,9191)
9191 FORMAT(///, ' NONEQUILIBRIUM RADIATION MODELS', ///)
IF (IQ4.EQ.1) CALL OLRAD (IQ4,IQ5,IQ6,IQ8,IQ9,IQ10)
IF (IQ4.EQ.2) CALL CARRAD (IQ4,IQ5,IQ9)
IF (IQ4.EQ.3) CALL CORRAD (IQ4,IQ5,IQ6,IQ8,IQ9)
IF (IQ4.EQ.4) CALL ANDRAD (IQ4,IQ5,IQ9)
IF ((IQ4.EQ.5).OR.(IQ4.EQ.6)) THEN
CALL OLRAD (IQ4,IQ5,IQ6,IQ8,IQ9,IQ10)
CALL CARRAD (IQ4,IQ5,IQ9)
CALL CORRAD (IQ4,IQ5,IQ6,IQ8,IQ9)
CALL ANDRAD (IQ4,IQ5,IQ9)
ENDIF
C
IF (IQ5.EQ.0) THEN
DO 1920 K = 3,NXCON
WRITE(8,1911) XPST(K)
DO 1920 J = 1,NPTS(K)
WRITE(8,1921) J,ALPN(K,J),ALPO(K,J),BETN(K,J),BETO(K,J)
1921 * FORMAT(' J = ',I2,' An = ',D11.4,' Ao = ',D11.4/
',2X,' BN = ',D11.4,' BO = ',D11.4)
1920 CONTINUE
C
ENDIF
DO 1912 K = 3,NXCON
WRITE(8,1911) XPST(K)
1911 FORMAT(///, ' AT X = ',D11.4,///)
DO 1912 J = 1,NPTS(K)
WRITE(8,1913) J,TEMP(K,J),RRHO(K,J),TEMPO(K,J)
1913 FORMAT(' J=',I2,' TE=',F10.2,' RHO=',D11.4,' T=',F10.2)
1912 CONTINUE
DO 1918 K = 3,NXCON
WRITE(8,1911) XPST(K)
DO 1918 J = 1,NPTS(K)
WRITE(8,1914) J
1914 FORMAT(/, ' AT J = ',I2,' NUMBER DENSITIES ARE ',/)
DO 1918 I = 1,IMAX
WRITE(8,1915) I,RCON(K,I,J)
1915 FORMAT(' SPECIES = ',I2,' N = ',D11.4)
1918 CONTINUE
WRITE(8,835)
845 WRITE(8,850)
850 FORMAT(1X,'END THIS CASE , HALLELUJAH '/')
REWIND 8
REWIND 14
STOP
END

```

```

C
C-----
C
SUBROUTINE BASIC
C
C BASIC CALLED BY CALITH TO EVALUATE DERIVATIVES H DOT,
C SUB I DOT AND EV SUB I DOT ; DERIVATIVES START IN DER(1)
C
IMPLICIT REAL*8(A-H,O-Z)
COMMON /A1/ TE1,TE2,M,IX,IMAX,IPSI,MODEL
COMMON /A2/ P(500),DPDX(500),VARI(500)
COMMON /A3/ EVIINF(25),THETAI(25),MUI(25),FI(25),DELHI(25),
* CIINF(25),LI(25)
COMMON /A4/ R,MUINF,DELTA,LAMSQ,OMEGSQ,OMEGA,MU,T,U,TINF
COMMON /A5/ EPSIIL(20,25),GIL(20,25),CID(200,50)
COMMON /A6/ VAR(52),CUVAR(52),DER(51)
COMMON /A7/ MJ(50),EJ(50),AJ(50),BJ(50),DIRECT(50)
COMMON /A8/ TVI(25),DGENI(25),BI(25),NI(25)
COMMON /A10/ NUIJ(26,50),AIJ(25,50),NUIJ(25,50)

```

```

COMMON /A11/ SIGIK(25,25),ALPIK(25,25),BETAIK(25,25)
COMMON /A13/ SP,TS,DELX,ZSTERM,IZTERM,NSR,MW
COMMON /A14/ EINF,PINF,RHOINF,VINF,E,UMAX,KEYINT,RHO,HSTAG
COMMON /A15/ PFTL,KITR1,NIP(25),UP(25)
COMMON /A16/ WE(25),WEXE(25),WEYE(25),WEZE(25),IUNEG
COMMON /A18/ ITNEG,IEXP
DIMENSION PHI(50),SJ(50),DCIDJDX(50),EVP(100)
DIMENSION CI(25),EVI(25),DCIDX(25),DEVIDX(25),EVIBAR(25)
EQUIVALENCE (CUVAR(2),H),(CUVAR(3),CI(1))
EQUIVALENCE (DER(1),DHDX),(DER(2),DCIDX(1))
REAL*8 KPJ,KEQ,KJ(50),LAMSQ,MU,MUI,MUIDT,MUINF,ND

C
C      DER(1) = DHDX          MUST BE DHDX AS H MAY BE + OR -
C      DER(2) = DCIDX(1)
C
C      DER(1+IMAX)          = DCIDX(IMAX)
C      DER(1+IMAX+1)        = DEVIDX(1)
C      DER(1+IMAX+IMAX) = DEVIDX(IMAX)
C
C      CUVAR(1) = X
C      CUVAR(2) = H,ENTHALPY - IT MUST BE H WHICH MAY BE + OR -
C      CUVAR(3) = CI(1)
C
C      CUVAR(2+IMAX)        = CI(IMAX)
C      CUVAR(2+IMAX+1)      = EVI(1)
C      CUVAR(2+IMAX+IMAX) = EVI(IMAX)
C
C                                     KEYINT = 1 AT END OF 1ST INTERVAL
C      IF (ITNEG.EQ.1.OR.IEXP.EQ.1) RETURN
C      IF (M.NE.O) GO TO 10
C      DO 5 I = 1,IMAX
C      J1 = I + IMAX + 2
C      5 CUVAR(J1) = VAR(J1)
C                                     INTERPOLATE FOR P ACROSS INTEGRATION INTERVAL
C      10 DO 15 I = 1,IMAX
C      15 EVI(I) = CUVAR(2+IMAX+I)
C      MFTL = 1
C      NFTL = IZTERM - IPSI + 1.
C      CALL FTLUP (CUVAR(1),PFTL,MFTL,NFTL,VARI(IPSI),P(IPSI))
C                                     INTERPOLATE FOR DPDX
C      CALL FTLUP (CUVAR(1),DPFTL,MFTL,NFTL,VARI(IPSI),DPDX(IPSI))
C      IF (KEYINT.EQ.O) GO TO 95
C      SUM = 0.0
C      DO 45 I = 1,IMAX
C      USE SHOCK VALUES FOR INITIAL COMPUTATION ON EACH STREAMLINE
C      SUM = SUM + CI(I)/MUI(I)
C      IF (EVI(I).EQ.O.O) TVI(I) = 0.0
C      IF (EVI(I).EQ.O.O) GO TO 45
C      XL = ((DGENI(I)*R*THETAI(I))/(MUI(I)*EVI(I)) + 1.0)
C      ALN = DLOG(XL)
C                                     COMPUTE TVI
C      TVI(I) = THETAI(I)/ALN
C      45 CONTINUE
C      MU = 1.0/SUM
C      U2 = 2.0*(HSTAG - H)
C      IF (U2.LT.O.O) WRITE(15,50) H
C      50 FORMAT(1X,'U**2 IS NEG IN BASIC , H = ',E11.4,2X,
C      * 'END STREAMLINE INTEGRATION')
C      IF (U2.LT.O.O) IUNEG = 1
C      IF (U2.LT.O.O) RETURN
C      U = DSQRT(U2)
C                                     ITERATE FOR E
C      KCODE = 0
C      55 KITR1 = 0
C      ICODE = 0
C      MAXI = 50
C      TOL1 = .001
C      TOL2 = .00001
C
C      CALL FOFE (MAXI,TOL1,TOL2,ICODE,E)
C
C      IF (ICODE.EQ.O) GO TO 75
C      IF (ICODE.EQ.1) WRITE(15,60)
C      60 FORMAT(1X,'* MAXIMUM ITERATION EXCEEDED IN SUB BASIC *: STOP 66'/)
C      IF (ICODE.EQ.2) WRITE(15,65)
C      65 FORMAT(1X,'*** DERIVATIVE = 0 IN SUB BASIC **** : STOP 66'/)
C      WRITE(15,70) ICODE
C      70 FORMAT(1X,'ICODE = ',I2,' IN SUB BASIC '/')
C      IF (ICODE.NE.1) STOP
C      IF (KCODE.EQ.3) STOP

```

```

C      WHEN ICODE = 1, TRY A NEW STARTING E. DO THIS 2 TIMES ;      STOP 66
      KCODE = KCODE + 1
      GO TO 55
75 IF (IEXP.NE.1) GO TO 85
      KCODE = KCODE + 1
      IEXP = 0
      IF (KCODE.EQ.1) GO TO 55
      IF (KCODE.GT.2) RETURN
      E = H - H*1.E-05
      GO TO 55
85 RHO = PFTL/(H - E)
      T = (PFTL*MU)/(RHO*R)
      IF (T.GT.0.0) GO TO 95
      WRITE(15,90) T,RHO,PFTL,H,E,MU
90 FORMAT(1X,'T NEGATIVE = ',E12.4,1X,'RHO = ',E11.4,1X,'PFTL = ',
*      E11.4,1X,'H = ',E11.4,1X,'E = ',E11.4,1X,'MU = ',E11.4,2X,
*      'IN SUB BASIC'/)
      E = H - H*1.0E-06
      ITNEG = 1
      RETURN
95 KEYINT = 1
      PI = 4.0*ATAN(1.0)
      BC = 1.38054E-16
      AVGN = 6.02252E+23

C      COMPUTE DCIDX AND DEVIDX FOR EACH SPECIE I
      DO 210 I = 1,IMAX
      DCIDX(I) = 0.0
      CVSUM1 = 0.0
      CVSUM2 = 0.0

C      CORRECTED RELAXATION TIME FOR PARK MODEL
      IF (MODEL.EQ.4.AND.I.EQ.1) THEN
      ND = RHO*AVGN*CI(I)/MUI(I)
      SIGMAV = 1.00E-16*(50000./T)**2
      C = DSQRT(8.*AVGN*BC*T/(PI*MUI(I)))
      TAUC = 1.0/(C*SIGMAV*ND)
      ENDIF

C      FOR EACH REACTION J
      DO 190 J = 1,JMAX
      ICT = MOD(J,2)

C      II IS THE SELECTED SPECIE FOR REACTION J
      KNT = 0
      IF (ICT.EQ.1) THEN
      IF (DIRECT(J).EQ.1.0) PHIC = 1.0
      IF (DIRECT(J).EQ.2.0) PHIC = 2.0
      II = MJ(J)
100 IF (M.EQ.0) GO TO 115
      IF (FI(II).NE.0.0) GO TO 120
115 IF (PHIC.EQ.1.0) PHI(J) = 1.0
      IF (PHIC.EQ.2.0) PHI(J+1) = 1.0
      GO TO 125
120 IF (MODEL.EQ.3.AND.UP(II).NE.0.0) GO TO 122
      TEM = DEXP(THETAI(II)/TVI(II))
      TEM1 = DEXP(THETAI(II)/T)
      TEM2 = DEXP(-NI(II)*(THETAI(II)/TVI(II) - THETAI(II)/T))
      TEM3 = DEXP(THETAI(II)/TVI(II) - THETAI(II)/T)
      IF (TEM3.EQ.1.0.AND.PHIC.EQ.1.0) PHI(J) = 1.0
      IF (TEM3.EQ.1.0.AND.PHIC.EQ.2.0) PHI(J+1) = 1.0
      IF (TEM3.EQ.1.0) GO TO 125

C      PHI = COUPLING COEFF.; FOR CVD,CVDV AND PARK MODEL
      IF (PHIC.EQ.1.0) PHI(J) = ((1.0-TEM2)/(TEM3-1.0)*(TEM - 1.0)
*      /(TEM1 - 1.0))/NI(II)
      IF (PHIC.EQ.2.0) PHI(J+1) = ((1.0-TEM2)/(TEM3-1.0)*(TEM - 1.0)
*      /(TEM1 - 1.0))/NI(II)
      GO TO 125
122 USUM = 0.0
      TSUM = 0.0
      TFSUM = 0.0
      TVTSUM = 0.0
      TF = 1.0/(1./TVI(II) - 1./T - 1./UP(II))
      NIC = NIP(II) + 1
      DO 123 N = 1,NIC
      XN = N
      EEVP = XN - 0.5
      EVP(N) = EEVP*(WE(II) + EEVP*(-WEXE(II) + EEVP*(WEYE(II) +
*      WEZE(II)*EEVP)))
      EVP(N) = EVP(N) - EVP(1)
      USUM = USUM + DEXP(EVP(N)/UP(II))
      TSUM = TSUM + DEXP(-EVP(N)/T)
      TFSUM = TFSUM + DEXP(-EVP(N)/TF)
123 TVTSUM = TVTSUM + DEXP(-EVP(N)/TVI(II))

```

```

C                                     COUPLING COEFF. FOR CVDV-PREFERENTIAL MODEL
IF (PHIC.EQ.1.0) PHI(J) = (TSUM*TFSUM)/(USUM*TVTSUM)
IF (PHIC.EQ.2.0) PHI(J+1) = (TSUM*TFSUM)/(USUM*TVTSUM)
125 KNT = KNT + 1
IF (KNT.EQ.1.AND.PHIC.EQ.1.0) THEN
  PHIC = 2.0
  II = MJ(J+1)
  GO TO 100
ENDIF
IF (KNT.EQ.1.AND.PHIC.EQ.2.0) THEN
  PHIC = 1.0
  II = MJ(J+1)
  GO TO 100
ENDIF

C                                     START OF LOOP TO DETERMINE EQUILIBRIUM CONSTANT
BETAJ = 0.0
FJORT = 0.0
DO 130 K = 1,IMAX
  SUMG = 0.0
  LII = LI(K)
  DO 128 L = 1,LII
128 SUMG = SUMG + GIL(L,K)*DEXP(-EPSIL(L,K)/T)
C                                     MUIOT = (CHEMICAL POTENTIAL)/T
MUIOT = - (BI(K) + (5. + 2.*FI(K))/2.*(DLOG(TINF) +
* DLOG(T/TINF)) + FI(K)*DLOG(1./((1. - DEXP(-THETAI(K)/T)))
* + DLOG(SUMG/GIL(1,K))) + DELHI(K)/(R*T)
BETAJ = BETAJ + (NUPIJ(K,J) - NUIJ(K,J))
130 FJORT = FJORT + (NUPIJ(K,J) - NUIJ(K,J))*MUIOT
C                                     KPJ = EQUIL. CONST. (IN TERMS OF PARTIAL PRESSURES)
KPJ = DEXP(-FJORT)
C                                     KEQ = EQUILIBRIUM CONSTANT
KEQ = KPJ*(9.8688225E-07*R*T)**(-BETAJ)
C                                     KJ = RATE CONSTANT (FORW: J = 1,3,.. BACK: J = 2,4,..)
IF (DIRECT(J).EQ.1.0) THEN
  KJ(J) = AJ(J)*T**BJ(J)*DEXP(-EJ(J)/T)
  IF (J.EQ.19) KJ(J) = AJ(J)*TE1**BJ(J)*DEXP(-EJ(J)/TE1)
  IF (J.EQ.21) KJ(J) = AJ(J)*TE1**BJ(J)*DEXP(-EJ(J)/TE1)
  IF (TE1.LT.2.D3) GO TO 131
  IF (J.EQ.19) KEQ = (1.8D-08)*(TE1**1.5D0)*DEXP(-1.69D5/TE1)
  IF (J.EQ.21) KEQ = (3.59D-09)*(TE1**1.5D0)*DEXP(-1.575D5/TE1)
131 KJ(J+1) = KJ(J)/KEQ
ELSE
  KJ(J+1) = AJ(J)*T**BJ(J)*DEXP(-EJ(J)/T)
  IF (J.EQ.19) KJ(J+1) = AJ(J)*TE1**BJ(J)*DEXP(-EJ(J)/TE1)
  IF (J.EQ.21) KJ(J+1) = AJ(J)*TE1**BJ(J)*DEXP(-EJ(J)/TE1)
  IF (TE1.LT.2.D3) GO TO 132
  IF (J.EQ.19) KEQ = (1.8D-08)*(TE1**1.5D0)*DEXP(-1.69D5/TE1)
  IF (J.EQ.21) KEQ = (3.59D-09)*(TE1**1.5D0)*DEXP(-1.575D5/TE1)
132 KJ(J) = KJ(J+1)*KEQ
ENDIF
ENDIF
IF (NUIJ(IMAX+1,J).EQ.0) SJ(J) = 1.0
IF (NUIJ(IMAX+1,J).EQ.0) GO TO 140
SUM = 0.0
DO 135 ISUM = 1,IMAX
135 SUM = SUM + AIJ(ISUM,J)*CI(ISUM)/MUI(ISUM)
SJ(J) = RHO*SUM
140 PROD = 1.0
DO 180 IPROD = 1,IMAX
  IF (NUIJ(IPROD,J).EQ.0) GO TO 180
  TEM = RHO*CI(IPROD)/MUI(IPROD)
  IF (TEM.GE.0.0) GO TO 165
C LET PROBLEM COMPUTE PROD. WHEN CI NEG ; DECIDE IN MAIN ABOUT ACCEPTING NEG CI
LABW = MOD(NUIJ(IPROD,J),2)
C                                     LABW = 0 FOR EVEN NUIJ, ABW = 1 FOR ODD NUIJ
IF (LABW.EQ.0) PROD = PROD*(-TEM)**NUIJ(IPROD,J)
IF (LABW.EQ.1) PROD = -1.0*PROD*(-TEM)**NUIJ(IPROD,J)
GO TO 180
165 PROD = PROD*TEM**NUIJ(IPROD,J)
180 CONTINUE
IF (ICT.EQ.1) PRODF = PROD
BETAIJ = NUPIJ(I,J) - NUIJ(I,J)
DCIJDJ(J) = PHI(J)*KJ(J)*SJ(J)*(MUI(I)/(RHO*U))*PROD*BETAIJ
C                                     DCIDJ = RATE OF PRODUCTION OF THE CONCENTRATION OF SPECIE I
DCIDJ(I) = DCIDJ(I) + DCIJDJ(J)
IF (ICT.EQ.0.AND.MODEL.GT.1) THEN
C                                     QIJ = NET RATE OF PRODUCTION
QIJ = DCIJDJ(J-1) + DCIJDJ(J)
DENOM = KEQ*PHI(J-1)*SJ(J-1)*PRODF
IF (DENOM.EQ.0.0) GO TO 190

```



```

C                               CHI = DEGREE OF NON-EQUILIBRIUM
      CHI = 1.0 - (PHI(J)*SJ(J)*PROD/DENOM)
      DENOM1 = CI(I)*CHI
      IF (DENOM1.EQ.0.0) GO TO 190
      CVSUM1 = CVSUM1 + QIJ/DENOM1
      CVSUM2 = CVSUM2 + QIJ*(1.0 - CHI)/DENOM1
      ENDIF
190 CONTINUE
C
      TEM = DEXP(THETAI(I)/T)
      TEM1 = (FI(I)*DGENI(I)*R*THETAI(I))/MUI(I)
      EVIBAR(I) = TEM1/(TEM - 1.0)
      IF (M.NE.0) GO TO 195
      VAR(I+IMAX*2) = EVIBAR(I)
      DEVIDX(I) = 0.0
      GO TO 210
195 TAUSUM = 0.0
      IF (FI(I).EQ.0.0) GO TO 210
C                               CVD MODEL
      DO 200 K = 1,IMAX
      TEM3 = DEXP(-THETAI(I)/T)
      TEM4 = DEXP(SIGIK(K,I)*T**(-1./3.))
      TEM5 = (FI(I)*ALPIK(K,I))/PFTL
      TAU1K = TEM5*(T**BETA1K(K,I)*TEM4)/(1.0 - TEM3)
C                               MILLIKAN & WHITE RELAXATION DATA
      IF (MW.EQ.1) THEN
      IF (I.EQ.1) TAU1K = 1.9E-05*DEXP(216.45*T**(-1./3.))/PFTL
      IF (I.EQ.8) TAU1K = 3.1845E-05*DEXP(198.66*T**(-1./3.))/PFTL
      ENDIF
      TEM6 = CI(K)*(EVIBAR(I) - EVI(I))
C                               PARK MODEL
      IF (MODEL.EQ.4.AND.I.EQ.1) THEN
      TAULP = TAU1K + TAUC
      PF = DABS((T - TVI(I))/(TS - TINF))** (SP - 1.)
      TAUSUM = TAUSUM + (TEM6/(TAULP*U))*PF
      GO TO 200
      ENDIF
      TAUSUM = TAUSUM + TEM6/(TAU1K*U)
200 CONTINUE
      IF (MODEL.EQ.1) GO TO 207
      IF (MODEL.EQ.3) GO TO 202
C                               ADDITIONAL TERMS FOR CVDV & PARK MODEL
      IF (I.EQ.5.OR.I.EQ.6.OR.I.EQ.8) GO TO 207
      TEM7 = DEXP(THETAI(I)/TVI(I) - THETAI(I)/T)
      TEM75 = DEXP(NI(I)/4.*(THETAI(I)/TVI(I) - THETAI(I)/T))
      TEM8 = TEM75**4
      CVDVT = ((THETAI(I)/(TEM7 - 1.) - NI(I)*THETAI(I)/(TEM8 - 1.))
      * R/MUI(I) - EVI(I))*CVSUM1 - ((0.5*(NI(I) - 1.)*THETAI(I)
      * R/MUI(I)) - EVI(I))*CVSUM2
      TAUSUM = TAUSUM + CVDVT
      GO TO 207
C                               ADDITIONAL TERMS FOR CVDV-PREFERENTIAL MODEL
202 IF (I.EQ.5.OR.I.EQ.6.OR.I.EQ.8) GO TO 207
      USUM1 = 0.0
      USUM2 = 0.0
      TFSUM1 = 0.0
      TFSUM2 = 0.0
      NIC = NIP(I) + 1
      TF = 1.0/(1./TVI(I) - 1./T - 1./UP(I))
      DO 204 N = 1,NIC
      XN = N
      EEVP = XN - 0.5
      EVP(N) = EEVP*(WE(I) + EEVP*(-WE(XE(I) + EEVP*(WEYE(I) +
      * WEZE(I)*EEVP)))
      EVP(N) = EVP(N) - EVP(1)
      USUM1 = USUM1 + DEXP(EVP(N)/UP(I))
      USUM2 = USUM2 + EVP(N)*DEXP(EVP(N)/UP(I))
      TFSUM1 = TFSUM1 + DEXP(-EVP(N)/TF)
204 TFSUM2 = TFSUM2 + EVP(N)*DEXP(-EVP(N)/TF)
      GBAR = (USUM2/USUM1)*(R/MUI(I))
      EBAR = (TFSUM2/TFSUM1)*(R/MUI(I))
      CVDVPT = (EBAR - EVI(I))*CVSUM1 - (GBAR - EVI(I))*CVSUM2
      TAUSUM = TAUSUM + CVDVPT
C                               DEVIDX = EQUILIBRIUM VIBRATIONAL ENERGY
207 DEVIDX(I) = TAUSUM
      DER(1+IMAX+I) = DEVIDX(I)
210 CONTINUE
C                               COMPUTE DHDX
      DHDX = DPFTL/RHO
      RETURN

```

```

C-----
C
SUBROUTINE FOFE (MAXI,TOL1,TOL2,ICODE,E1)
C
CALLED BY SUB BASIC TO EVALUATE E BY NEWTON ITERATION TECHNIQUE
C
IMPLICIT REAL*8(A-H,O-Z)
COMMON /A1/ TE1,TE2,M,IX,IMAX,IPSI,MODEL
COMMON /A2/ P(500),DPDX(500),VARI(500)
COMMON /A3/ EVIINF(25),THETAI(25),MUI(25),FI(25),DELHI(25),
* CIINF(25),LI(25)
COMMON /A4/ R,MUINF,DELTA,LAMSQ,OMEGSQ,OMEGA,MU,T,U,TINF
COMMON /A5/ EPSIIL(20,25),GIL(20,25),CID(200,50)
COMMON /A6/ VAR(52),CUVAR(52),DER(51)
COMMON /A7/ MJ(50),EJ(50),AJ(50),BJ(50),DIRECT(50)
COMMON /A8/ TVI(25),DGENI(25),BI(25),NI(25)
COMMON /A10/ NUIJ(26,50),AIJ(25,50),NUPIJ(25,50)
COMMON /A15/ PFTL,KITR1,NIP(25),UP(25)
COMMON /A18/ ITNEG,IEXP
DIMENSION CI(25)
EQUIVALENCE (CUVAR(3),CI(1)),(CUVAR(2),H)
REAL*8 MUI,LAMSQ,MUINF,MU

IF (IEXP.EQ.1.OR.ITNEG.EQ.1) RETURN
KITR1 = KITR1 + 1
ITER = 0
PORHO = H - E1
T = PORHO*MU/R
1 ITER = ITER + 1
E = 0.0
DE = 0.0
DO 35 I = 1,IMAX
IF (TVI(I).EQ.0.0) THEN
TEM1 = (1.5*R*T)/MUI(I) + (FI(I)*R*T)/MUI(I)
DTEM1 = 1.5*R/MUI(I) + FI(I)*R/MUI(I)
GO TO 15
ENDIF
TEM = DEXP(THETAI(I)/TVI(I))
TEM1 = (1.5*R*T)/MUI(I) + (FI(I)*R*T)/MUI(I) +
* (FI(I)*R*THETAI(I))/(MUI(I)*(TEM - 1.0))
DTEM1 = 1.5*R/MUI(I) + FI(I)*R/MUI(I) + (R*FI(I)*THETAI(I)**2
* / (MUI(I)*TVI(I)**2)*(TEM/(TEM - 1.0)**2)
15 SUMG = 0.0
SUMGE = 0.0
DSUMGE = 0.0
LII = LI(I)
C
DO 25 L = 1,LII
TEM3 = - EPSIIL(L,I)/T
IF (TEM3.LT.741.67) GO TO 22
IEXP = 1
RETURN
22 TEM2 = DEXP(TEM3)
SUMG = SUMG + GIL(L,I)*TEM2
SUMGE = SUMGE + GIL(L,I)*EPSIIL(L,I)*TEM2
IF (SUMGE.LT.1.0D+34) GO TO 25
IEXP = 1
RETURN
25 DSUMGE = DSUMGE + TEM2*GIL(L,I)*EPSIIL(L,I)**2
DEVI = R/MUI(I)*(((SUMG*DSUMGE/T**2) - (SUMGE/T)**2)
* /SUMG**2)
DEI = DTEM1 + DEVI
EI = TEM1 + R/MUI(I)*(SUMGE/SUMG) + DELHI(I)/MUI(I)
DE = DE + DEI*CI(I)
E = E + EI*CI(I)
35 CONTINUE
C
IF (E.GT.H) ITNEG = 1
DFT = - DE
FT = E1 - E
IF (DFT.EQ.0.0) THEN
ICODE = 2
RETURN
ENDIF
TS1 = T - (FT/DFT)
DELT = DABS(TS1 - T)
T = TS1
E1 = H - R*T/MU

```

```

IF (ITER.GE.MAXI) THEN
  ICODE = 1
  RETURN
ENDIF
IF (DELT.GT.TOL1) THEN
  ITER1 = ITER
  GO TO 1
ENDIF
ITER2 = ITER1 + 10
IF (DELT.LE.TOL2.OR.ITER.GE.ITER2) RETURN
GO TO 1
END

```

```

C
C-----
C
SUBROUTINE FOFTS1 (MAXI,TOL1,TOL2,ICODE,TSG)
C
C CALLED BY MAIN TO EVALUATE TS BY NEWTON ITERATION TECHNIQUE
C
  IMPLICIT REAL*8(A-H,O-Z)
  COMMON /A1/ TE1,TE2,M,IX,IMAX,IPSI,MODEL
  COMMON /A2/ P(500),DPDX(500),VARI(500)
  COMMON /A3/ EVIINF(25),THETAI(25),MUI(25),FI(25),DELHI(25)
  *      ,CIINF(25),LI(25)
  COMMON /A4/ R,MUINF,DELTA,LAMSQ,OMEGSQ,OMEGA,MU,T,U,TINF
  COMMON /A5/ EPSIIL(20,25),GIL(20,25),CID(200,50)
  COMMON /A6/ VAR(52),CUVAR(52),DER(51)
  REAL*8 MUI,LAMSQ,MUINF,MU

  ITER = 0
1  ITER = ITER + 1
  ES = 0.0
  DES = 0.0
  DO 25 I = 1,IMAX
    IF (M.EQ.1) EVIS = EVIINF(I)
    IF (M.EQ.1) GO TO 5
    TEM = DEXP(THETAI(I)/TSG)
    EVIS = (R*THETAI(I))/(MUI(I)*(TEM-1.0))*FI(I)
    DEVIS = (R*FI(I)*THETAI(I)**2)/(MUI(I)*TSG**2)*(TEM/(TEM-1.0)**2)
  5  SUMG = 0.0
    SUMGE = 0.0
    DSUMGE = 0.0
    LII = LI(I)

  C
    DO 10 L = 1,LII
      TEM1 = DEXP(- EPSIIL(L,I)/TSG)
      SUMG = SUMG + TEM1*GIL(L,I)
      SUMGE = SUMGE + TEM1*GIL(L,I)*EPSIIL(L,I)
  10  DSUMGE = DSUMGE + TEM1*GIL(L,I)*EPSIIL(L,I)**2
      EEIS = (R/MUI(I))*(SUMGE/SUMG)
      DEEIS = R/MUI(I)*(((SUMG*DSUMGE/TSG**2) - (SUMGE/TSG)**2)
      *      /SUMG**2)
      EIS = (1.5*R*TSG)/MUI(I) + (FI(I)*R*TSG)/MUI(I) + EVIS + EEIS
      *      + DELHI(I)/MUI(I)
      DEIS = 1.5*R/MUI(I) + FI(I)*R/MUI(I) + DEVIS + DEEIS
      ES = ES + EIS*CIINF(I)
      DES = DES + DEIS*CIINF(I)
  25  CONTINUE

  C
    FAC = DSQRT(OMEGSQ - 2.0*LAMSQ*(DELTA - ES))
    FIN = (2.0*MUINF*(DELTA - ES)*FAC)/(R*(OMEGA + FAC))
    FTS = TSG - FIN
    FAC1 = R*(OMEGA+FAC)*(MUINF*(DELTA-ES)*(2.0/FAC)*LAMSQ*DES -
    *      2.0*FAC*MUINF*DES) - 2.0*MUINF*(DELTA-ES)*R*LAMSQ*DES
    FAC2 = R**2*(OMEGA + FAC)**2
    DFTS = 1.0 - FAC1/FAC2
    IF (DFTS.EQ.0.0) THEN
      ICODE = 2
      RETURN
    ENDIF
    TSG1 = TSG - (FTS/DFTS)
    DELTS = DABS(TSG1 - TSG)
    TSG = TSG1
    IF (ITER.GE.MAXI) THEN
      ICODE = 1
      RETURN
    ENDIF
    IF (DELTS.GT.TOL1) THEN
      ITER1 = ITER
      GO TO 1
    ENDIF
  
```

```

ENDIF
ITER2 = ITER1 + 10
IF (DELTS.LE.TOL2.OR.ITER.GE.ITER2) RETURN
GO TO 1
END

```

SUBROUTINE CHECK

CHECK CALLED BY CALITH TO MAKE DECISION TO ACCEPT ANSWERS

```

IF ACCEPTABLE      ; SET ELB = 0 AND RETURN
IF NOT ACCEPTABLE ; MODIFY SPEC AND CI ; SET ELB = 1.0 AND RETURN

```

```

IMPLICIT REAL*8(A-H,O-Z)
COMMON /A1/ TE1,TE2,M,IX,IMAX,IPSI,MODEL
COMMON /A2/ P(500),DPDX(500),VARI(500)
COMMON /A3/ EVIINF(25),THETAI(25),MUI(25),FI(25),DELHI(25),
*      CIINF(25),LI(25)
COMMON /A4/ R,MUINF,DELTA,LAMSQ,OMEGSQ,OMEGA,MU,T,U,TINF
COMMON /A5/ EPSIIL(20,25),GIL(20,25),CID(200,50)
COMMON /A6/ VAR(52),CUVAR(52),DER(51)
COMMON /A13/ SP,TS,DELX,ZSTERM,IZTERM,NSR,MW
COMMON /A17/ ELB,SPEC,CJ,TPREV,HPREV,HCHECK,TCHECK
COMMON /A18/ ITNEG,IEXP
REAL*8 MUI,MU,MUINF,LAMSQ
EQUIVALENCE (CUVAR(2),H)

```

```

IF (IEXP.NE.1) GO TO 1
GO TO 15

```

REDUCE COMPUTING INTERVAL TO TRY AND AVOID EXP ERROR STOP

```

1 IF (ITNEG.EQ.0) GO TO 4

```

```

E.GT.H SOMETIMES LEADS TO NEG T--REDUCE INTERVAL TO TRY TO AVOID INSTABILITY
GO TO 15

```

```

4 IMAXP1 = IMAX + 1
DO 10 I = 3,IMAXP1

```

CUVAR(2) = H MAY BE NEGATIVE

```

IF (CUVAR(1).LT.0.0) WRITE(15,5) I,IX,CUVAR(1)

```

```

IF (CUVAR(1).LT.0.0) GO TO 15

```

```

5 FORMAT(1X,'NEG CI AT I = ',I3,1X,'IX = ',I3,1X,'CUVAR(1) = ',
*      E12.5,1X,'IN SUB CHECK'/)

```

```

10 CONTINUE

```

```

IF (IX.LT.3) GO TO 30

```

```

IF ((IPSI.EQ.(NSR+1)).AND.(IX.LT.(NSR+3))) GO TO 30

```

```

IF (DABS(TPREV - T)/T.LT.TCHECK) GO TO 25

```

```

WRITE(15,11)

```

```

11 FORMAT(1X,'CHECK ABS(TPREV-T)/T.GE.TCHECKT IN SUB CHECK'/)

```

```

15 IF (SPEC.GT.1.0E-15) GO TO 20

```

```

WRITE(15,*)'SPEC .LT. 1.0E-15 IN SUB CHECK
STOP

```

STOP 30

REDUCE INTERVAL

```

20 SPEC = SPEC/4.0

```

```

CJ = SPEC

```

```

ELB = 1.0

```

```

WRITE(15,22) SPEC,IPSI,IX,T,TPREV,H,HPREV,IEXP,ITNEG

```

```

22 FORMAT(1X,'REDUCED SPEC = ',E9.4,2X,'IPSI = ',I3,2X,'IX = ',I3,

```

```
*      2X,'T = ',E9.4,2X,'TPREV = ',E9.4,/,1X,'H = ',E9.4,3X,

```

```
*      'HPREV = ',E9.4,3X,'IEXP = ',I3,3X,'ITNEG = ',I3,' IN CHECK'/)

```

```

ITNEG = 0

```

```

IEXP = 0

```

```

RETURN

```

```

25 IF (DABS((HPREV - H)/H).GT.HCHECK) WRITE(15,27)

```

```

27 FORMAT(1X,'CHECK ABS((HPREV-H)/H).GT.HCHECKT IN SUB CHECK'/)

```

```

IF (DABS((HPREV - H)/H).GT.HCHECK) GO TO 15

```

ACCEPTABLE

```

30 TPREV = T

```

```

HPREV = H

```

```

ELB = 0.0

```

```

RETURN

```

```

END

```

SUBROUTINE SHOCKG (NXPSTM,ITK)

SUBROUTINE SHOCKG CALLED BY MAIN AND MUST BE SUPPLIED BY USER.

X,ZS,RS,RC,COST,SINT FOR EACH X FROM 0.0 TO X AT ZSTERM IN INCREMENTS

OF (DELX/NSR) TO DELX AND INCREMENTS OF DELX THEREAFTER.

```

C      X      =      DISTANCE ALONG SHOCK
C      ZS      =      DISTANCE ALONG SHOCK AXIS OF SYMMETRY
C      RS      =      RADIUS OF SHOCK
C      RC      =      RADIUS OF CURVATURE OF SHOCK
C      COST     =      COS OF ANGLE OF ATTACK
C      SINT     =      SIN OF ANGLE OF ATTACK
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /A9/ EVIS(25),XPST(100)
      COMMON /A12/ SINTM(1500),COSTM(1500),RSM(1500),RCM(1500),X1(1500)
      *          ,ZSM(1500)
      COMMON /A13/ SP,TS,DELX,ZSTERM,IZTERM,NSR,MW
      DIMENSION ZDUM(501),RSDUM(501),RCDUM(501),XC(501),CODUM(501),
      *          SIDUM(501)
C      RS/L = SQRT(2*R*Zs/L - Bs*(Zs/L)**2)      WHERE: L = 1.0 cm
      ITK = 0
      EL = 1.0
      R = 230.0
      BS = -4.0
      DO 5 I = 2,501
5      ZDUM(I) = (I-1)*ZSTERM/500
      ZDUM(1) = 0.0
      RSDUM(1) = 0.0
      XC(1) = 0.0
      CODUM(1) = 0.0
      SIDUM(1) = 1.0
      RCDUM(1) = 230.0
C
      DO 10 I = 2,501
      ZSND = ZDUM(I)/EL
      RSDUM(I) = DSQRT(2.*R*ZSND - BS*ZSND**2)
      C = 1.0/(R - BS*ZSND)**2
      C1 = DSQRT(C*RSDUM(I)**2 + 1.0)
      XC(I) = RSDUM(I)/2.*C1 + 1./(2.*DSQRT(C))*DLOG(RSDUM(I)*DSQRT(C)
      *      + C1)
      D1 = (R - BS*ZSND)/RSDUM(I)
      D2 = -(R - BS*ZSND)**2/RSDUM(I)**3 - BS/RSDUM(I)
      THETA = ATAN(D1)
      SIDUM(I) = DSIN(THETA)
      CODUM(I) = DCOS(THETA)
      RCDUM(I) = (1.0 + D1**2)**1.5/DABS(D2)
C
C      DIMENSIONALIZED QUANTITIES
      RCDUM(I) = EL*RCDUM(I)
      XC(I) = EL*XC(I)
      RSDUM(I) = EL*RSDUM(I)
10      CONTINUE
C
      WRITE(8,25)
25      FORMAT(5X,'X',11X,'ZS',10X,'RS',10X,'RC',9X,'COST',8X,'SINT'/)
      NOX = 0
      M = 1
      NZS = 501
C
C      INTERPOLATE SHOCK VALUES TO X SHOCK COORDINATE
30      NOX = NOX + 1
      IF (NOX.LT.(NSR+2)) X = X + DELX/NSR
      IF (NOX.EQ.1) X = 0.0
      IF (NOX.GT.(NSR+1)) X = X + DELX
      CALL FTLUP (X,ZS,M,NZS,XC,ZDUM)
      CALL FTLUP (X,RS,M,NZS,XC,RSDUM)
      CALL FTLUP (X,RC,M,NZS,XC,RCDUM)
      CALL FTLUP (X,COST,M,NZS,XC,CODUM)
      CALL FTLUP (X,SINT,M,NZS,XC,SIDUM)
      WRITE(8,35) X,ZS,RS,RC,COST,SINT,NOX
35      FORMAT(1X,6(E10.4,2X),I4)
      SINTM(NOX) = SINT
      COSTM(NOX) = COST
      RSM(NOX) = RS
      RCM(NOX) = RC
      X1(NOX) = X
      ZSM(NOX) = ZS
      DO 36 IT = 2,NXPSTM
      IF (DABS(XPST(IT) - X).GT.1.0E-06) GO TO 36
      ITK = ITK + 1
36      CONTINUE
      IF (ZS.GE.0.0) GO TO 40
      WRITE(15,37) ZS
37      FORMAT(1X,'ZS SHOULD BE GREATER THAN 0.0 : ZS = ',E10.5,
      *      /,1X,'STOP 13'/)
      STOP

```

```

C
40 IF (ZS.LT.ZSTERM) GO TO 30
   IZTERM = NOX
   RETURN
   END

C
-----
C
SUBROUTINE CALITH (N,CIMAX,PHMAX)

C
IN THE CALITH VERSION OF CALINT, THE VARIABLE IN VAR(2) AND
C
CUVAR(2) MAY BE + OR -. VALUES OF OTHER DEPENDENT VARIABLES
C
ARE EXPECTED TO BE POSITIVE. 10-69
C
IMPLICIT REAL*8(A-H,O-Z)
COMMON /A1/ TE1,TE2,M,IX,IMAX,IPSI,MODEL
COMMON /A2/ P(500),DPDX(500),VARI(500)
COMMON /A3/ EVIINF(25),THETA(25),MU(25),FI(25),DELHI(25),
*      CIINF(25),LI(25)
COMMON /A4/ R,MUINF,DELTA,LAMSG,OMEGSQ,OMEGA,MU,T,U,TINF
COMMON /A5/ EPSIIL(20,25),GIL(20,25),CID(200,50)
COMMON /A6/ VAR(52),CUVAR(52),DER(51)
COMMON /A7/ MJ(50),EJ(50),AJ(50),BJ(50),DIRECT(50)
COMMON /A8/ TVI(25),DGENI(25),BI(25),NI(25)
COMMON /A10/ NUIJ(26,50),AIJ(25,50),NUIJ(25,50)
COMMON /A11/ SIGIK(25,25),ALPIK(25,25),BETAIK(25,25)
COMMON /A17/ ELB,SPEC,CJ,TPREV,HPREV,HCHECK,TCHECK
COMMON /A19/ ELE1(51),ELE2(51),NERR
REAL*8 MU,MUINF,LAMSG
DIMENSION F1(51),F2(51),F3(51),CAPF1(51),CAPF2(51),CAPF3(51)
*      ,P1(51),PH(51),DELT(51),Y3(51),Y4(51),F4(51),Y2(51)

C
C IF CJ OR N IS EQUAL TO ZERO : NERR = 1
C IF ELE1 IS LESS THAN OR EQUAL TO ELE2 : NERR = 2
C
TEST INPUT
FN = N
TEST = CJ*FN
IF (TEST) 998,997,998
997 NERR = 1
RETURN
998 DO 999 I = 1,N
   IF ((ELE1(I) - ELE2(I)).LE.O.O) THEN
     NERR = 2
     RETURN
   ENDIF
999 CONTINUE
1000 IF (SPEC) 5,1,5
C
SECTION FOR INITIALIZATION COMPUTATION OF DERIVATIVES
1 SPEC = CJ
  ICONT = 1
2 N1 = N + 1
  DO 3 I = 1,N1
3 CUVAR(I) = VAR(I)
  CALL BASIC
C
RETURN WITH DERIVATIVES IN DER
DO 4 I = 1,N
4 F1(I) = DER(I)
RETURN
C
COMPUTE Y2,X2
5 CUVAR(1) = VAR(1) + CJ/2.0
  DO 6 I = 1,N
  I1 = I + 1
  Y2(I) = VAR(I1) + CJ/2.0*F1(I)
  IF (I.EQ.1) GO TO 6
  IF (Y2(I)) 65,6,6
6 CUVAR(I1) = Y2(I)
  GO TO 66
65 SPEC = CJ
  CJ = CJ/2.0
  IF (CJ.LE.1.0E-25) THEN
    WRITE(15,25) CJ
25 FORMAT(1X,'MIN. INTEGRATION STEP OF ',E12.4,' WAS REACHED')
    STOP
  ENDIF
  GO TO 5
C
CALL BASIC TO EVALUATE F2
66 CALL BASIC
C
RETURN
DO 7 I = 1,N
I1 = I + 1

```

```

      F2(I) = DER(I)
C
      Y3(I) = VAR(I1) + CJ/2.0*F2(I)      COMPUTE Y3
      IF (I.EQ.1) GO TO 7
      IF (Y3(I)) 65,7,7
7    CUVAR(I1) = Y3(I)
C
      CALL BASIC                          CALL BASIC TO EVALUATE F3
C
      DO 10 I = 1,N                      RETURN
      F3(I) = DER(I)
C
      COMPUTE P,PH AND CAP F TERMS
      IF (Y3(I) - Y2(I)) 9,8,9
8    P1(I) = 0.0
      GO TO 91
9    P1(I) = -((F3(I) - F2(I))/(Y3(I) - Y2(I)))
91   PH(I) = P1(I)*CJ
      IF (PH(I)) 83,83,103
83   PH(I) = 0.0
      P1(I) = 0.0
      GO TO 84
103  Z1 = DABS(Y3(I) - Y2(I))/((DABS(Y3(I)) + DABS(Y2(I)))/2.0)
      IF (Z1 - 0.5E-04) 83,83,84
84   IF (PH(I) - 0.1) 85,85,95
85   CAPF1(I) = 1.0 - PH(I)/2.0 + PH(I)**2/6.0 - PH(I)**3/24.0
      CAPF2(I) = 0.5 - PH(I)/6.0 + PH(I)**2/24.0 - PH(I)**3/120.0
      CAPF3(I) = 1./6. - PH(I)/24. + PH(I)**2/120. - PH(I)**3/720.
      GO TO 10
95   CAPF1(I) = (DEXP(-PH(I)) - 1.0)/(- PH(I))
      CAPF2(I) = (CAPF1(I) - 1.0)/(- PH(I))
      CAPF3(I) = (CAPF2(I) - 0.5)/(- PH(I))
10   CONTINUE
C
      IS PH BETWEEN ELE2 AND ELE1
      IF (ICONT - 1) 101,101,102
102  ICONT = ICONT - 1
      SPEC = CJ
      GO TO 17
101  DO 11 I = 1,N
      IF (DABS(PH(I)) - ELE1(I)) 11,11,13
11   CONTINUE
      SPEC = CJ
      GO TO 15
C
      HALVE INTERVAL AND DOUBLE PH RANGE
13   DO 96 I = 1,N
      ELE1(I) = ELE1(I)*2.0
      IF (ELE1(I) - PHMAX) 94,94,955
94   ELE2(I) = ELE2(I)*2.0
      GO TO 96
955  ELE1(I) = ELE1(I)/2.0
96   CONTINUE
      SPEC = CJ
      CJ = CJ/2.0
      ICONT = 3
      GO TO 5
C
      RETURN TO RECOMPUTE INTERVAL
15   DO 16 I = 1,N
      IF (DABS(PH(I)) - ELE2(I)) 16,17,17
16   CONTINUE
C
      DOUBLE INTERVAL
      CJ = 2.0*CJ
      IF (CJ - CIMAX) 17,17,165
165  CJ = CIMAX
C
      COMPUTE Y4,X4
17   DO 18 I = 1,N
      I1 = I + 1
      CUVAR(I1) = VAR(I1) + SPEC*(F3(I)*(2.0*CAPF2(I)) + F1(I)*
      * (CAPF1(I) - 2.0*CAPF2(I)) + F2(I)*PH(I)*CAPF2(I))
      IF (I.EQ.1) GO TO 18
      IF (CUVAR(I1)) 175,18,18
175  CJ = SPEC
      CJ = CJ/2.0
      GO TO 5
18   Y4(I) = CUVAR(I1)
      CUVAR(1) = VAR(1) + SPEC
C
      CALL BASIC                          CALL BASIC TO EVALUATE F4
C
      DO 20 I = 1,N                      RETURN
      I1 = I + 1
      F4(I) = DER(I)

```

```

C                                     COMPUTE DELTA Y
C
  DELTY(I) = SPEC*(F1(I)*CAPF1(I)+(-3.0*(F1(I)+P1(I)*VAR(I1))+2.0*
* (F2(I)+P1(I)*Y2(I)) + 2.0*(F3(I)+P1(I)*Y3(I)) - F4(I) - P1(I)*
* Y4(I))*CAPF2(I) + 4.0*((F1(I)+P1(I)*VAR(I1)) - (F2(I)+P1(I)*
* Y2(I)) - (F3(I)+P1(I)*Y3(I)) + (F4(I)+P1(I)*Y4(I)))*CAPF3(I))
C
C                                     COMPUTE Y + DELTA Y
20  CUVAR(I1) = VAR(I1) + DELTY(I)
C                                     CALL CHECK FOR DECISION TO ACCEPT OR RECOMPUTE INTERVAL
  CALL CHECK
  IF (ELB) 21,21,23
C
C                                     UPDATE Y VALUES
21  N1 = N + 1
  DO 22 I = 2,N1
    I1 = I - 1
22  VAR(I) = VAR(I) + DELTY(I1)
  VAR(1) = VAR(1) + SPEC
C                                     RETURN TO COMPUTE DERIVATIVES AT Y + DELTA Y
  GO TO 2
C                                     RETURN TO RECOMPUTE INTERVAL
23  GO TO 5
  END
C
C-----
C
  SUBROUTINE FTLUP (X,Y,M,N,VARI,VARD)
C
C  THIS SUBROUTINE IS A MODIFICATION OF LIBRARY INTERPOLATION
C  SUBROUTINE FTLUP  REVISED 7-7-69
C
  IMPLICIT REAL*8(A-H,O-Z)
  DIMENSION VARI(5),VARD(5),V(3),YY(2),II(100)
C  INITIALIZE ALL INTERVAL POINTERS TO -1.0 FOR MONOTONICITY CHECK
  DO 4 J = 1,100
4  II(J) = -1
  MA = IABS(M)
C  ASSIGN INTERVAL POINTER FOR GIVEN VARI TABLE ; THE SAME POINTER
C  WILL BE USED ON A GIVEN VARI TABLE EVERY TIME
  LOCF = VARI(1)
  LI = MOD(LOCF,100) + 1
  I = II(LI)
  IF (I.GE.0) GO TO 10
  IF (N.LT.2) GO TO 10
C
C  MONOTONICITY CHECK
  IF (VARI(2) - VARI(1)) 1,1,3
C  ERROR IN MONOTONICITY
2  K = LOCF
  WRITE(15,102) J,K
102 FORMAT(1X,'TABLE BELOW OUT OF ORDER FOR FTLUP AT POSITION ',
* I5,/,1X,'X TABLE IS STORED IN LOCATION',I5/)
  DO 103 J = 1,N
103 WRITE(15,*) VARI(J),VARD(J)
  STOP
C
C  MONOTONIC DECREASING
1  DO 5 J = 2,N
  IF (VARI(J) - VARI(J-1)) 5,2,2
5  CONTINUE
  GO TO 10
C
C  MONOTONIC INCREASING
3  DO 6 J = 2,N
  IF (VARI(J) - VARI(J-1)) 2,2,6
6  CONTINUE
C
C  INTERPOLATION
10 IF (I.LE.0) I = 1
  IF (I.GE.N) I = N - 1
  IF (N.LE.1) GO TO 8
  IF (MA.NE.0) GO TO 99
C
C  ZERO ORDER
8  Y = VARD(1)
  GO TO 800
C
C  LOCATE I INTERVAL (X(I).LE.X.LT.X(I+1))
99 IF ((VARI(I) - X)*(VARI(I+1) - X)) 61,61,40
C  IN GIVES DIRECTION FOR SEARCH INTERVALS
40 SIGN1 = 1.0
  IN = SIGN(SIGN1,(VARI(I+1) - VARI(I))*(X - VARI(I)))
C  IF X OUTSIDE ENDPOINTS, EXTRAPOLATE FROM END INTERVAL
41 IF ((I+IN).LE.0) GO TO 61
  IF ((I+IN).GE.N) GO TO 61
  I = I + IN

```



```

        IF ((VARI(I) - X)*(VARI(I+1) - X)) 61,61,41
61  IF (MA.EQ.2) GO TO 200
C      FIRST ORDER
      Y = (VARD(I)*(VARI(I+1) - X) - VARD(I+1)*(VARI(I) - X))/
      * (VARI(I+1) - VARI(I))
      GO TO 800
C      SECOND ORDER
200  IF (N.EQ.2) GO TO 2
      IF (I.EQ.(N-1)) GO TO 209
      IF (I.EQ.1) GO TO 201
C      PICK THIRD POINT
      SK = VARI(I+1) - VARI(I)
      IF ((SK*(X - VARI(I-1))).LT.(SK*(VARI(I+2) - X))) GO TO 209
201  L = I
      GO TO 702
209  L = I - 1
702  V(1) = VARI(L) - X
      V(2) = VARI(L+1) - X
      V(3) = VARI(L+2) - X
      YY(1) = (VARD(L)*V(2)-VARD(L+1)*V(1))/(VARI(L+1)-VARI(L))
      YY(2) = (VARD(L+1)*V(3)-VARD(L+2)*V(2))/(VARI(L+2)-VARI(L+1))
      Y = (YY(1)*V(3) - YY(2)*V(1))/(VARI(L+2) - VARI(L))
800  II(LI) = I
      RETURN
      END

```

```

C-----
C
C      SUBROUTINE FOFT23 (MAXI,TOL1,TOL2,ICODE,TSG,IQ1,IQ2,IQ3)
C
C      CALLED BY MAIN TO EVALUATE TS BY NEWTON ITERATION TECHNIQUE
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /A1/ TE1,TE2,M,IX,IMAX,IPSI,MODEL
      COMMON /A2/ P(500),DPDX(500),VARI(500)
      COMMON /A3/ EVIINF(25),THETAI(25),MUI(25),FI(25),DELHI(25),
      * CIINF(25),LI(25)
      COMMON /A4/ R,MUINF,DELTA,LAM SQ,OMEGSQ,OMEGA,MU,T,U,TINF
      COMMON /A5/ EPSIIL(20,25),GIL(20,25),CID(200,50)
      COMMON /A6/ VAR(52),CUVAR(52),DER(51)
      DIMENSION SUMGM(25),EISM(25),DEISM(25)
      REAL*8 MUI,LAM SQ,MUINF,MU
C
      W = 1.0
      DO 1 I = 1,IMAX
1    CID(IX,I) = CIINF(I)
      ITER = 0
2    ITER = ITER + 1
      IF (ITER.EQ.1) THEN
        ES = DELTA
        PS = DSQRT(OMEGSQ - 2.*LAM SQ*(DELTA - ES))
      ENDIF
      IF (IQ1.EQ.1) THEN
        TVIB = TINF
      ELSE
        TVIB = TSG
      ENDIF
      IF (IQ2.EQ.1) THEN
        TEL = TINF
      ELSE
        TEL = TSG
      ENDIF
C
      ES = 0.0
      DES = 0.0
      DO 25 I = 1,IMAX
      IF (M.EQ.1) EVIS = EVIINF(I)
      IF (M.EQ.1) GO TO 5
      TEM = DEXP(THETAI(I)/TVIB)
      EVIS = (R*THETAI(I))/(MUI(I)*(TEM - 1.0))*FI(I)
      DEVIS = (R*FI(I)*THETAI(I)**2)/(MUI(I)*TVIB**2)*(TEM/(TEM-1.0)**2)
5    IF (I.EQ.1) QVN = 1.0/(1.0 - DEXP(-THETAI(I)/TVIB))
      IF (I.EQ.2) QVO = 1.0/(1.0 - DEXP(-THETAI(I)/TVIB))
      SUMGM(I) = 0.0
      SUMGE = 0.0
      DSUMGE = 0.0
      LII = LI(I)
C
      DO 10 L = 1,LII
      TEM1 = DEXP(-EPSIIL(L,I)/TEL)

```

```

SUMGM(I) = SUMGM(I) + TEM1*GIL(L,I)
SUMGE = SUMGE + TEM1*GIL(L,I)*EPSIIL(L,I)
10 DSUMGE = DSUMGE + TEM1*GIL(L,I)*EPSIIL(L,I)**2
QROTN = TSG/5.8
QROTO = TSG/4.2
EEIS = (R/MUI(I))*(SUMGE/SUMGM(I))
DEEIS = R/MUI(I)*(((SUMGM(I)*DSUMGE/TEL**2) - (SUMGE/TEL)**2)
* /SUMGM(I)**2)
EISM(I) = (1.5*R*TSG)/MUI(I) + (FI(I)*R*TSG)/MUI(I) + EVIS + EEIS
* + DELHI(I)/MUI(I)
DEISM(I) = 1.5*R/MUI(I) + FI(I)*R/MUI(I) + DEVIS + DEEIS
25 CONTINUE
C BETA FOR OXYGEN
IF (IQ3.EQ.2) THEN
  TD = 59500.0
  F1 = CIINF(2)*MUINF/MUI(2)
  C1A = TD/TSG
  C1A = DEXP(C1A/2.0)
  C1B = 8.8568195*(1. - CIINF(1))*C1A*(1./(TSG**2.5))
  C1 = C1A*2.0*MUINF*PS*C1B*QVO*QROTO*(SUMGM(2)/(SUMGM(4)**2))/R
  BETAO = ((F1-1.) + DSQRT((1.-F1)**2 + 4.*(F1+C1)))/(2.*(F1+C1))
  BETAN = 0.0
ENDIF
C BETA FOR NITROGEN AND OXYGEN
IF (IQ3.EQ.3) THEN
  BETAN = .1
  BETAO = 1.0
  TDN = 113500.0
  TDO = 59500.0
  F1A = 8.8568195*(1. - CIINF(1))*DEXP(TDO/TSG)*(1./(TSG**2.5))
  F2A = 12.36681*(1. - CIINF(2))*DEXP(TDN/TSG)*(1./(TSG**2.5))
  F1 = 2.0*MUINF*PS*F1A*QVO*QROTO*(SUMGM(2)/(SUMGM(4)**2))/R
  F2 = 2.0*MUINF*PS*F2A*QVN*QROTN*(SUMGM(1)/(SUMGM(3)**2))/R
30 A = ((1. - BETAO**2) + BETAN*(1. - BETAO))/(BETAO**2) - F1
  B = ((1. - BETAN**2) + BETAO*(1. - BETAN))/(BETAN**2) - F2
  ABO = (BETAN*BETAO - 2.*(1. + BETAN))/(BETAO**3)
  ABN = (1. - BETAO)/(BETAO**2)
  BBO = (1. - BETAN)/(BETAN**2)
  BBN = (BETAN*BETAO - 2.*(1. + BETAO))/(BETAN**3)
  DEN = ABO*BBN - BBO*ABN
  DBO = (-A*BBN + B*ABN)/DEN
  DBN = (-B*ABO + A*BBO)/DEN
  BETAN = BETAN + DBN
  BETAO = BETAO + DBO
  IF ((DABS(DBO).GT..00001).OR.(DABS(DBN).GT..00001)) GO TO 30
ENDIF
C DISSOCIATING CONCENTRATIONS ACROSS THE SHOCK
CID(IX,2) = (1.0 - BETAO)*CIINF(2)
CID(IX,4) = BETAO*CIINF(2)
CID(IX,1) = (1.0 - BETAN)*CIINF(1)
CID(IX,3) = BETAN*CIINF(1)
C
DO 35 I = 1,IMAX
ES = ES + EISM(I)*CID(IX,I)
35 DES = DES + DEISM(I)*CID(IX,I)
FAC = DSQRT(OMEGSQ - 2.0*LAMSQ*(DELTA - ES))
FIN = (2.0*MUINF*(DELTA - ES)*FAC)/(R*(OMEGA + FAC))
FTS = TSG - FIN
FAC1 = R*(OMEGA+FAC)*(MUINF*(DELTA-ES)*(2.0/FAC)*LAMSQ*DES -
* 2.0*FAC*MUINF*DES) - 2.0*MUINF*(DELTA-ES)*R*LAMSQ*DES
FAC2 = R**2*(OMEGA + FAC)**2
DFTS = 1.0 - (FAC1/FAC2)
PS = FAC
IF (DFTS.EQ.0.0) THEN
  ICODE = 2
  RETURN
ENDIF
TSG1 = TSG - W*(FTS/DFTS)
DELTS = DABS(TSG1 - TSG)
TSG = TSG1
IF (ITER.GE.MAXI) THEN
  ICODE = 1
  RETURN
ENDIF
IF (DELTS.GT.TOL1) THEN
  ITER1 = ITER
  GO TO 2
ENDIF
ITER2 = ITER1 + 10
IF (DELTS.LE.TOL2.OR.ITER.GE.ITER2) RETURN

```

GO TO 2
END

C
C
C

SUBROUTINE OLRAD (IQ4,IQ5,IQ6,IQ8,IQ9,IQ10)
IMPLICIT REAL*8(A-H,O-Z)
COMMON /A1/ TE1,TE2,M,IX,IMAX,IPSI,MODEL
COMMON /A3/ EVIINF(25),THETAI(25),MUI(25),FI(25),DELHI(25),
* CIINF(25),LI(25)
COMMON /A5/ EPSIIL(20,25),GIL(20,25),CID(200,50)
COMMON /A9/ EVIS(25),XPST(100)
COMMON /RAD/ RCON(100,10,40),RRHO(100,40),TEMP(100,40)
* ,TSTAG,Y(100,40),NXCON,NPTS(100)
COMMON /RADC/ YBDY(100),B(100,8,40),OPTL(100,8,40)
COMMON /RADD/ TEMPO(100,40),ALPN(100,40),ALPD(100,40)
COMMON /RADOL/ BETN(100,40),BETO(100,40)
REAL*8 KAPPA(100,8,40),KAP(100,4,40),KA(100,4,40),MUI
REAL*8 K11,K22,N,NN
WRITE(8,1)

1 FORMAT(///' OLSTAD RADIATION MODEL')
IF(IQ8.EQ.0) THEN
IF(IQ6.EQ.0) WRITE(8,2)
IF(IQ6.EQ.1) WRITE(8,3)
ENDIF
2 FORMAT('/' BETA=BETA(Te)')
3 FORMAT('/' BETA=BETA(Tt)')

C

NUMBER DENSITIES

DO 12 K=3,NXCON
DO 12 J=1,NPTS(K)
DO 10 II=1,IMAX
XXX=RRHO(K,J)*6.023D+23
XXX=XXX*RCON(K,II,J)/MUI(II)
10 RCON(K,II,J)=XXX
12 CONTINUE
IF(IQ10.EQ.0)GO TO 15
DO 14 K=3,NXCON
NUP=NPTS(K)-1
DO 14 J=1,NUP
T=TEMPO(K,J)
FNA=RCON(K,3,J)
FNI=RCON(K,9,J)
FNE=RCON(K,7,J)
FNM=RCON(K,1,J)
FOA=RCON(K,4,J)
FOI=RCON(K,10,J)
FOM=RCON(K,2,J)

C

ELECTRON TEMPERATURE MODEL

ITRTE=0
EI=2.3322D-11
EO=2.1735D-11
FAV=6.023D23
SA=1.52D-15
F1=0
F2=0
FO=0
574 ITRTE=ITRTE+1
IF(ITRTE.GT.1)GO TO 571
KK=1
569 KK=KK+1
IF(KK.GT.50)TE1=2000
IF(KK.GT.50)GO TO 14
TE=1000*KK
571 SM=0.5355D-19*TE+.696D-15
IF(ITRTE.GT.50)GO TO 573
SI=1.53614D08*(TE**3)/FNE
SI=DSQRT(SI)
SI=((4.38384D-06)/(TE**2))*DLOG(SI)
FKF=4.16D13*(TE**(.050D0))*DEXP(-1.20D05/TE)
FK1=5.49D13*(TE**(.050D0))*DEXP(-1.045D05/TE)
FKB=2.3D21*(TE**(-1.0D00))*DEXP(4.9D4/TE)
FK5=1.53D22*(TE**(-1.0D00))*DEXP(5.3D4/TE)
WEA=FKF*FNA-FKB*(FNI/FAV)*FNE
W2E=FK1*FOA-FK5*(FOI/FAV)*FNE
SX=(FNA+FOA)*SA+(FNI+FOI)*SI+(FNM+FOM)/(2.0D0)*SM
W1=WEA+W2E
C W2=DER(8)*U*FAV/VAR(9)
C W1=W2
FE=T-1.23357D-10/SX/DSQRT(TE)*(WEA*EI+W2E*EO+W1*3.45D-16*TE)-TE
IF(ITRTE.GT.1)GO TO 568

```

      IF((FE/DABS(FE)*FO).LT.O.DOO)GO TO 567
      FO=FE
      TO=TE
      GO TO 569
567 F1=FE
      T1=TE
566 TE=(TO*F1-T1*FO)/(F1-FO)
      IF(DABS(TE-T1).LT.1.DOO)GO TO 573
      IF(DABS(TE-TO).LT.1.DOO)GO TO 573
      GO TO 574
568 IF((FE/DABS(FE)*FO/DABS(FO)).LT.O.OOOO)GO TO 564
      TO=TE
      FO=FE
      GO TO 566
564 F1=FE
      T1=TE
      GO TO 566
C 573 CONTINUE
573 IF(TE.GT.T)TE=T
      IF(ITRTE.GT.50)WRITE(6,572)
572 FORMAT(' ','TE DID NOT CONVERGE')
      TEMP(K,J)=TE
      14 CONTINUE
      15 CONTINUE
C
C      NONEQUILIBRIUM RADIATION CORRECTION
      IF (IQ5.EQ.O) THEN
      DO 122 K=3,NXCON
      DO 122 J=1,NPTS(K)
      IF(J.EQ.NPTS(K)) THEN
      ALPN(K,J)=O.D+OO
      BETN(K,J)=O.D+OO
      ALPO(K,J)=O.D+OO
      BETO(K,J)=O.D+OO
      GO TO 122
      ENDIF
      Q1N=O.D+O
      Q2N=O.D+O
      Q3N=O.D+O
      DO 112 III=1,LI(3)
112 Q1N=Q1N+GIL(III,3)*DEXP(-EPSIL(III,3)/TEMP(K,J))
      DO 113 III=1,LI(9)
113 Q2N=Q2N+GIL(III,9)*DEXP(-EPSIL(III,9)/TEMP(K,J))
      DO 130 III=1,LI(1)
130 Q3N=Q3N+GIL(III,1)*DEXP(-EPSIL(III,1)/TEMP(K,J))
      RH01=RCON(K,3,J)*1.401D+O1/6.023D+23
      RH02=RCON(K,9,J)*1.401D+O1/6.023D+23
      RH03=RCON(K,1,J)*2.802D+O1/6.023D+23
      RH04N=RH01+RH02+RH03
      QVIBN=1.D+O/((1.D+O-DEXP(-3.39D+O3/TEMP(K,J)))
      QROTN=TEMP(K,J)/5.8D+O
      Q1O=O.D+O
      Q2O=O.D+O
      Q3O=O.D+O
      DO 152 III=1,LI(4)
152 Q1O=Q1O+GIL(III,4)*EXP(-EPSIL(III,4)/TEMP(K,J))
      DO 153 III=1,LI(10)
153 Q2O=Q2O+GIL(III,10)*DEXP(-EPSIL(III,10)/TEMP(K,J))
      DO 160 III=1,LI(2)
160 Q3O=Q3O+GIL(III,2)*DEXP(-EPSIL(III,2)/TEMP(K,J))
      RH01=RCON(K,4,J)*1.600D+O1/6.023D+23
      RH02=RCON(K,10,J)*1.600D+O1/6.023D+23
      RH03=RCON(K,2,J)*3.200D+O1/6.023D+23
      RH04O=RH01+RH02+RH03
      QVIBO=1.D+O/((1.D+O-DEXP(-2.27D+O3/TEMP(K,J)))
      QROTO=TEMP(K,J)/4.2D+O
      C1=RH04N*2.4701D+O1*QVIBN*QROTN*Q3N*DEXP(1.135D+O5/TEMP(K,J))
      C1=C1/(Q1N**2*TEMP(K,J)**1.5)
      C2=RH04O*1.7725D+O1*QVIBO*QROTO*Q3O*DEXP(5.950D+O4/TEMP(K,J))
      C2=C2/(Q1O**2*TEMP(K,J)**1.5)
      BETN(K,J)=(RCON(K,3,J)/RCON(K,1,J))*RCON(K,3,J)*C1*1.162D-23
      * /RH04N
      * ALPN(K,J)=RCON(K,9,J)/RCON(K,3,J)*RCON(K,7,J)*Q1N/Q2N*
      * DEXP(1.69D+O5/TEMP(K,J))/(4.826D+15*TEMP(K,J)**1.5)
      BETO(K,J)=(RCON(K,4,J)/RCON(K,2,J))*RCON(K,4,J)*C2*1.328D-23
      * /RH04O
      * ALPO(K,J)=RCON(K,10,J)/RCON(K,4,J)*RCON(K,7,J)*Q1O/Q2O*
      * DEXP(1.575D+O5/TEMP(K,J))/(4.826D+15*TEMP(K,J)**1.5)
122 CONTINUE
C
C      ABSORPTION COEFFICIENTS
      DO 20 K=3,NXCON

```

```

DD 20 J=1,NPTS(K)
TT=TEMP(K,J)/168800.0
IF(IQ6.EQ.1) THEN
  TM=TEMPO(K,J)/168800.0
ELSE
  TM=TT
ENDIF
C IF((IQ6.EQ.1).AND.(TEMPO(K,J).LE.8.D+03)) GO TO 20
C IF(TEMP(K,J).LE.8.D+03) GO TO 20
IF(IQ8.EQ.0) THEN
  O=BETO(K,J)
  N=BETN(K,J)
ELSE
  O=1.D+O
  N=1.D+O
ENDIF
KAPPA(K,4,J)=5.D-19*D*RCON(K,2,J)+5.D-20*N*RCON(K,1,J)
KAP(K,4,J)=1.7D-17*RCON(K,3,J)*DEXP(-.246/TT)
KA(K,4,J)=O.D+O
KAPPA(K,3,J)=2.D-18*(N*RCON(K,1,J)+O*RCON(K,2,J))+KAPPA(K,4,J)
KAP(K,3,J)=2.1D-17*RCON(K,3,J)*DEXP(-.165/TT)+KAP(K,4,J)
KA(K,3,J)=O.D+O
KAPPA(K,2,J)=5.1D-18*(N*RCON(K,1,J)+O*RCON(K,2,J))+KAPPA(K,3,J)
KAP(K,2,J)=KAP(K,3,J)
KA(K,2,J)=5.1D-18*RCON(K,4,J)
KAPPA(K,1,J)=2.D-17*D*RCON(K,2,J)+4.D-16*N*RCON(K,1,J)
C +KAPPA(K,2,J)
KAP(K,1,J)=1.1D-17*RCON(K,3,J)+KAP(K,2,J)
KA(K,1,J)=KA(K,2,J)
IF(IQ5.EQ.0) THEN
  N=ALPN(K,J)
  O=ALPO(K,J)
ELSE
  N=1.D+O
  O=1.D+O
  ALPN(K,J)=N
  ALPO(K,J)=O
ENDIF
IF(IQ8.EQ.0) THEN
  NN=BETN(K,J)
  OO=BETO(K,J)
ELSE
  NN=1.D+O
  OO=1.D+O
  BETN(K,J)=NN
  BETO(K,J)=OO
ENDIF
KAPPA(K,5,J)=7.7D-17*(NN*RCON(K,1,J)+OO*RCON(K,2,J))
C *DEXP(-.49/TM)+
C 2.6D-17*(RCON(K,3,J)*N+RCON(K,4,J)*O)*DEXP(-.723/TT)
KAPPA(K,6,J)=2.D-18*RCON(K,2,J)*OO+6.OO-18*(RCON(K,3,J)*N+
C RCON(K,4,J)*O)*DEXP(-.379/TT)+KAPPA(K,5,J)
IF(RCON(K,7,J).LE.1.D+OO) THEN
  KAPPA(K,7,J)=KAPPA(K,6,J)
  GO TO 909
ENDIF
KAPPA(K,7,J)=1.2D03*((RCON(K,3,J)*N+RCON(K,4,J)*O)/RCON(K,7,J))*
C DEXP(-.489/TT)+KAPPA(K,6,J)
909 KAPPA(K,8,J)=3.2D-17*(RCON(K,3,J)*N+RCON(K,4,J)*O)*DEXP(-.631/TT)
C +KAPPA(K,5,J)
20 CONTINUE

```

PLANCK FUNCTION

```

C DD 30 K=3,NXCON
DD 30 J=1,NPTS(K)
TT=TEMP(K,J)/168800.0
C IF(TEMP(K,J).LE.8.D+03) GO TO 30
BLACK=(TT*168800.)*4*5.6696D-12/3.1415927
FF=15.D+OO/(3.1415927**4)
X1=1./TT
B1=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C DEXP(-2.*X1)*O.5*(.75+1.5*X1+1.5*X1**2+X1**3))
X1=0.935/TT
B2=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C DEXP(-2.*X1)*O.5*(.75+1.5*X1+1.5*X1**2+X1**3))
X1=0.835/TT
B3=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C DEXP(-2.*X1)*O.5*(.75+1.5*X1+1.5*X1**2+X1**3))
X1=0.754/TT
B4=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C DEXP(-2.*X1)*O.5*(.75+1.5*X1+1.5*X1**2+X1**3))

```

```

X1=0.473/TT
B6=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C DEXP(-2.*X1)*0.5*(.75+1.5*X1+1.5*X1**2+X1**3))
X1=0.213/TT
B8=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C DEXP(-2.*X1)*0.5*(.75+1.5*X1+1.5*X1**2+X1**3))
B8=1-B8
B2=B2-B1
B3=B3-B2
B4=B4-B3
B6=B6-B4
B5=1-B8-B6-B4-B3-B2-B1
B(K,1,J)=BLACK*B1
B(K,2,J)=BLACK*B2
B(K,3,J)=BLACK*B3
B(K,4,J)=BLACK*B4
B(K,6,J)=BLACK*B6
B(K,7,J)=BLACK*2.4D-21*RCON(K,7,J)*B6*DEXP(.162/TT)
B(K,8,J)=BLACK*B8
B(K,5,J)=BLACK*B5
30 CONTINUE
C TAU'S
DO 100 K=3,NXCON
DO 100 I=1,8
IF(I.GT.4) THEN
OPTL(K,I,1)=KAPPA(K,I,1)*DABS(Y(K,1)-YBDY(K))
ELSE
OPTL(K,I,1)=(KAPPA(K,I,1)+KAP(K,I,1)+KA(K,I,1))
C *DABS(Y(K,1)-YBDY(K))
ENDIF
DO 100 J=2,NPTS(K)
IF(I.GT.4) THEN
K11=KAPPA(K,I,J)+KAPPA(K,I,J-1)
ELSE
K11=KAPPA(K,I,J)+KAP(K,I,J)+KAPPA(K,I,J-1)+KAP(K,I,J-1)
C +KA(K,I,J)+KA(K,I,J-1)
ENDIF
OPTL(K,I,J)=OPTL(K,I,J-1)+(5.D-1*(K11)*DABS(Y(K,J-1)
C -Y(K,J)))
100 CONTINUE
C QRW
DO 200 K=3,NXCON
WRITE(8,102) XPST(K)
102 FORMAT(///,' AT X = ',D11.4)
WQR=0.0
DO 180 I=1,8
CALL FEI(E1I,E2I,E3I,OPTL(K,I,1))
EOLD=E2I
JIN=1
IF(I.GT.4) THEN
K11=KAPPA(K,I,1)*B(K,I,1)
ELSE
IF(IQ5.EQ.1) THEN
ALPN(K,1)=1.D+O
ALPO(K,1)=1.D+O
ENDIF
K11=KAPPA(K,I,1)*B(K,I,1)+KAP(K,I,1)*B(K,I,1)*ALPN(K,1)+
C KA(K,I,1)*B(K,I,1)*ALPO(K,1)
ENDIF
SUM=3.1415927*EOLD*K11*DABS(Y(K,1)-YBDY(K))
IF(IQ9.EQ.0) WRITE(8,2221) JIN,I,SUM
DO 150 J=2,NPTS(K)
IF(I.GT.4) THEN
K11=KAPPA(K,I,J)*B(K,I,J)
K22=KAPPA(K,I,J-1)*B(K,I,J-1)
ELSE
IF(IQ5.EQ.1) THEN
ALPN(K,J)=1.D+O
ALPO(K,J)=1.D+O
ENDIF
K11=KAPPA(K,I,J)*B(K,I,J)+KAP(K,I,J)*B(K,I,J)*ALPN(K,J)+
C KA(K,I,J)*B(K,I,J)*ALPO(K,J)
K22=KAPPA(K,I,J-1)*B(K,I,J-1)+
C KAP(K,I,J-1)*B(K,I,J-1)*ALPN(K,J-1)+
C KA(K,I,J-1)*B(K,I,J-1)*ALPO(K,J-1)
ENDIF
Z=OPTL(K,I,J)
CALL FEI(E1I,E2I,E3I,Z)
STSUM=3.14159*(K11*E2I+K22*EOLD)*DABS(Y(K,J-1)-Y(K,J))
IF(IQ9.EQ.0) THEN

```

```

      WRITE(8,2221) J,I,STSUM
2221  FORMAT(1X,' AT J =',I2,' BAND ',I2,' QR =',D11.4)
      ENDIF
145  SUM=SUM+STSUM
      EOLD=E2I
150  CONTINUE
      WRITE(8,103) I,SUM
103  FORMAT(/,' FOR BAND ',I2,' QR = ',D11.4,' WATTS/SQ.CM.')
      WQR=WQR+SUM
180  CONTINUE
      WRITE(8,101) WQR
101  FORMAT(/' TOTAL QR = ',D11.4,' WATTS/SQ.CM.')
200  CONTINUE
      RETURN
      END

```

C
C
C

```

      SUBROUTINE FEI(E1I,E2I,E3I,Z)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /RADC/ YBDY(100),B(100,8,40),OPTL(100,8,40)
      CALL EXPI(Z,E1I,AUX)
      E2I=DEXP(-Z)-Z*E1I
      E3I=(DEXP(-Z)-Z*E2I)/2.0
      E2I=DEXP(-Z*DSQRT(3.D+00))
      RETURN
      END

```

C
C
C

```

      SUBROUTINE EXPI(X,RES,AUX)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /RADC/ YBDY(100),B(100,8,40),OPTL(100,8,40)
      IF(X-1.) 2,1,1
1    YY=1./X
      AUX=1.-YY*((((YY+3.377358)*YY+2.052156)*YY+2.709479D-01)/(((YY*
C1.072553+5.716943)*YY+6.945239)*YY+2.593888)*YY+2.709496D-01)
      RES=AUX*YY*DEXP(-X)
      RETURN
2    IF(X+3.) 6,6,3
3    AUX=(((((((7.122452D-7*X-1.766345D-6)*X+2.928433D-5)*X-2.335379D-4
C)*X+1.664156D-3)*X-1.041576D-2)*X+5.555682D-2)*X-2.500001D-1)*X
C+9.999999D-1
      RES=-1.D+30
      IF(X) 4,5,4
4    RES=X*AUX-DLOG(ABS(X))-5.772157D-1
5    RETURN
6    IF(X+9.) 8,8,7
7    AUX=1.-((((5.176245D-2*X+3.061037)*X+3.243655D+1)*X+2.244234D+2)*X
C+2.486697D+2)/((((X+3.995161)*X+3.893944D+1)*X+2.263818D+1)*X
C+1.807837D+2)
      GO TO 9
8    YY=9./X
      AUX=1.-YY*((((YY+7.659824D-1)*YY-7.271015D-1)*YY-1.080693)/(((YY
C*2.518750+1.122927D+1)*YY+5.921405)*YY-8.666702)*YY-9.724216)
9    RES=AUX*DEXP(-X)/X
      RETURN
      END

```

C
C
C

```

      SUBROUTINE CARRAD (IQ4,IQ5,IQ9)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /A1/ TE1,TE2,M,IX,IMAX,IPSI,MODEL
      COMMON /A3/ EVIINF(25),THETAI(25),MUI(25),FI(25),DELHI(25),
*      CIINF(25),LI(25)
      COMMON /A5/ EPSIIL(20,25),GIL(20,25),CID(200,50)
      COMMON /A9/ EVIS(25),XPST(100)
      COMMON /RAD/ RCON(100,10,40),RRHO(100,40),TEMP(100,40)
*      ,TSTAG,Y(100,40),NXCON,NPTS(100)
      COMMON /RADC/ YBDY(100),B(100,8,40),OPTL(100,8,40)
      COMMON /RADO/ TEMPO(100,40),ALPN(100,40),ALPO(100,40)
      REAL*8 KAPPA(100,8,40),MUI
      WRITE(8,1)

```

```

1  FORMAT(///' CARLSON RADIATION MODEL')
C      NUMBER DENSITIES
      IF((IQ4.EQ.5).OR.(IQ4.EQ.6)) GO TO 1111
      DO 12 K=3,NXCON
      DO 12 J=1,NPTS(K)
      DO 10 II=1,IMAX

```

```

10 RCON(K,II,J)=RRHO(K,J)*6.023D+23*RCON(K,II,J)/MUI(II)
12 CONTINUE
C      NONEQUILIBRIUM RADIATION CORRECTION
  IF (IQ5.EQ.O) THEN
    DO 122 K=3,NXCON
    DO 122 J=1,NPTS(K)
    IF(J.EQ.NPTS(K)) THEN
      ALPN(K,J)=O.D+OO
      GO TO 122
    ENDIF
    Q1=O.D+O
    Q2=O.D+O
    Q3=O.D+O
    DO 112 III=1,LI(3)
112  Q1=Q1+GIL(III,3)*DEXP(-EPSIIL(III,3)/TEMP(K,J))
    DO 113 III=1,LI(9)
113  Q2=Q2+GIL(III,9)*DEXP(-EPSIIL(III,9)/TEMP(K,J))
    DO 130 III=1,LI(1)
130  Q3=Q3+GIL(III,1)*DEXP(-EPSIIL(III,1)/TEMP(K,J))
    RH01=RCON(K,3,J)*1.401D+01/6.023D+23
    RH02=RCON(K,9,J)*1.401D+01/6.023D+23
    RH03=RCON(K,1,J)*2.802D+01/6.023D+23
    RH04=RH01+RH02+RH03
    QVIB=1.D+O/(1.D+O-DEXP(-3.39D+03/TEMP(K,J)))
    QROT=TEMP(K,J)/5.8D+O
    C1=RH04*2.4701D+01*QVIB*QROT*Q3*DEXP(1.135D+05/TEMP(K,J))
    C1=C1/(Q1**2*TEMP(K,J)**1.5)
    C2=RH04*8.909D+06*Q1*DEXP(1.69D+05/TEMP(K,J))
    C2=C2/(Q2*TEMP(K,J)**1.5)
    ALPHA=O.5D+O
    BETA=O.5D+O
132  F=BETA**2*(1.D+O-ALPHA)**2*C1+BETA-1.D+O
    FA=-2.D+O*BETA**2*(1.D+O-ALPHA)*C1
    FB=2.D+O*BETA*(1.D+O-ALPHA)**2*C1+1.D+O
    G=ALPHA**2*BETA*C2+ALPHA-1.D+O
    GA=2.D+O*ALPHA*BETA*C2+1.D+O
    GB=ALPHA**2*C2
    DENOM=FA*GB-FB*GA
    DA=(-F*GB+G*FB)/DENOM
    DB=(-G*FA+F*GA)/DENOM
    ALPHA=ALPHA+DA
    BETA=BETA+DB
    IF((DABS(DA).GT.1.D-05).OR.(DABS(DB).GT.1.D-5)) GO TO 132
    ALPH=RCON(K,9,J)/(RCON(K,9,J)+RCON(K,3,J))
    BET=(RCON(K,9,J)+RCON(K,3,J))/(RCON(K,9,J)+RCON(K,3,J)
    *      +2.D+O*RCON(K,1,J))
    *      ALPN(K,J)=(ALPH**2/(1.D+O-ALPH))*((1.D+O-ALPHA)/(ALPHA**2))
    *      *(BET/BETA)
122  CONTINUE
    ENDIF
1111 CONTINUE
C      PLANCK FUNCTION
  DO 13 K=3,NXCON
  DO 13 J=1,NPTS(K)
  TTT=TEMP(K,J)
  CALL PLANCK(B(K,1,J),B(K,2,J),B(K,3,J),B(K,4,J),B(K,5,J),TTT)
13  IF(IQ5.EQ.O) B(K,1,J)=ALPN(K,J)*B(K,1,J)
C      ABSORPTION COEFFICIENTS
  DO 20 K=3,NXCON
  DO 20 J=1,NPTS(K)
  DO 20 I=1,5
  KAPPA(K,I,J)=ABSORB(K,J,I)
  IF((IQ5.EQ.O).AND.(I.NE.1)) KAPPA(K,I,J)=KAPPA(K,I,J)
  *      *ALPN(K,J)
20  CONTINUE
C      TAU'S
  DO 100 K=3,NXCON
  DO 100 I=1,5
  OPTL(K,I,1)=KAPPA(K,I,1)*DABS(Y(K,1)-YBDY(K))
  DO 100 J=2,NPTS(K)
  OPTL(K,I,J)=OPTL(K,I,J-1)+(O.5*(KAPPA(K,I,J)+KAPPA(K,I,J-1))*
  C      DABS(Y(K,J-1)-Y(K,J)))
100  CONTINUE
C      QRW
  DO 200 K=3,NXCON
  WRITE(8,102) XPST(K)
102  FORMAT(//,' AT X = ',D11.4)
  WQR=O.O
  DO 180 I=1,5
  CALL FEI(E1I,E2I,E3I,OPTL(K,I,1))

```



```

EOLD=E2I
JIN=1
SUM=3.1415927*B(K,I,1)*EOLD*KAPPA(K,I,1)*DABS(Y(K,1)-YBDY(K))
IF(IQ9.EQ.O) WRITE(8,2221) JIN,I,SUM
DO 150 J=2,NPTS(K)
Z=OPTL(K,I,J)
IF(Z.EQ.O.D+O) THEN
E2I=1.D+O
GO TO 145
ENDIF
CALL FEI(E1I,E2I,E3I,Z)
STSUM=3.14159*(B(K,I,J)*KAPPA(K,I,J)*E2I+B(K,I,J-1)*
C KAPPA(K,I,J-1)*EOLD)*DABS(Y(K,J-1)-Y(K,J))
IF(IQ9.EQ.O) THEN
WRITE(8,2221) J,I,STSUM
2221 FORMAT(1X,' AT J =',I2,' BAND ',I2,' QR =',D11.4)
ENDIF
145 SUM=SUM+STSUM
EOLD=E2I
150 CONTINUE
WRITE(8,103) I,SUM
103 FORMAT(/' FOR BAND ',I2,' QR = ',D11.4,' WATTS/SQ.CM.')
WQR=WQR+SUM
180 CONTINUE
WRITE(8,101) WQR
101 FORMAT(/' TOTAL QR=',D11.4,' WATTS/SQ.CM.')
200 CONTINUE
RETURN
END

```

C

C

```

SUBROUTINE PLANCK(B1,B2,B3,B4,B5,YY)
IMPLICIT REAL*8(A-H,O-Z)
COMMON /RAD/ RCON(100,10,40),RRHD(100,40),TEMP(100,40)
* ,TSTAG,Y(100,40),NXCON,NPTS(100)
COMMON /RADC/ YBDY(100),B(100,8,40),OPTL(100,8,40)
T=YY
IF(T.LE.20000.O)GOTO1027
1027 IF(T.GE.8000.O)GOTO1029
B1=0.O
B2=0.O
B3=0.O
B4=0.O
B5=0.O
RETURN
1029 IF(T.LE.10000.O)GOTO1030
IF(T.LE.12000.O)GOTO1031
IF(T.LE.14000.O)GOTO1032
IF(T.LE.16000.O)GOTO1033
IF(T.LE.18000.O)GOTO1034
GOTO1240
1030 B1=10.O**(-65.26966+15.87146*DLOG10(T))
B2=10.O**(-54.11981+13.22561*DLOG10(T))
B3=10.O**(-44.83075+11.05391*DLOG10(T))
B4=10.O**(-16.52511+4.425444*DLOG10(T))
B5=10.O**(-8.874156+2.216357*DLOG10(T))
B1=B1*1000.OO
B2=B2*1000.OO
B3=B3*1000.OO
B4=B4*1000.OO
B5=B5*1000.OO
RETURN
1031 B1=(10.O**(-54.67501+13.22279*DLOG10(T)))*1000.O
B2=(10.O**(-44.60436+10.84674*DLOG10(T)))*1000.O
B3=(10.O**(-36.91518+9.075018*DLOG10(T)))*1000.O
B4=(10.O**(-15.30031+4.119244*DLOG10(T)))*1000.O
B5=(10.O**(-7.831680+1.955738*DLOG10(T)))*1000.O
RETURN
1032 B1=(10.O**(-47.32488+11.42093*DLOG10(T)))*1000.O
B2=(10.O**(-37.87564+9.197217*DLOG10(T)))*1000.O
B3=(10.O**(-31.30212+7.698991*DLOG10(T)))*1000.O
B4=(10.O**(-14.17392+3.843113*DLOG10(T)))*1000.O
B5=(10.O**(-7.128949+1.783465*DLOG10(T)))*1000.O
RETURN
1033 B1=(10.O**(-41.90448+10.11359*DLOG10(T)))*1000.O
B2=(10.O**(-32.87168+7.990316*DLOG10(T)))*1000.O
B3=(10.O**(-27.12177+6.690739*DLOG10(T)))*1000.O
B4=(10.O**(-13.12000+3.588921*DLOG10(T)))*1000.O
B5=(10.O**(-6.625956+1.662149*DLOG10(T)))*1000.O

```

```

      RETURN
1034 B1=(10.0**(-37.77432+9.131181*DLOG10(T)))*1000.0
      B2=(10.0**(-28.96032+7.059955*DLOG10(T)))*1000.0
      B3=(10.0**(-23.90936+5.926629*DLOG10(T)))*1000.0
      B4=(10.0**(-12.13764+3.355253*DLOG10(T)))*1000.0
      B5=(10.0**(-6.259400+1.574959*DLOG10(T)))*1000.0
      RETURN
1240 B1=(10.0**(-34.49941+8.36166*DLOG10(T)))*1000.0
      B2=(10.0**(-25.87330+6.33544*DLOG10(T)))*1000.0
      B3=(10.0**(-21.34066+5.32298*DLOG10(T)))*1000.0
      B4=(10.0**(-11.28150+3.1540*DLOG10(T)))*1000.0
      B5=(10.0**(-5.97387+1.50786*DLOG10(T)))*1000.0
      RETURN
      END

C
C-----
C
      FUNCTION ABSORB(K,I,ISTEP)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /RAD/ RCON(100,10,40),RRHD(100,40),TEMP(100,40)
      *      ,TSTAG,Y(100,40),NXCON,NPTS(100)
      COMMON /RADC/ YBDY(100),B(100,8,40),OPTL(100,8,40)
C      THIS PROGRAM USES 5 WAVELENGTH STEPS
C      THE NEXT CARD MUST BE CHANGED FOR GRAY GAS CASE
      IGRAY=0.0
      T=TEMP(K,I)
      BLACKB=((5.6697D-05)*(T**4))/3.14159
      YYY=T
      YY=T/10000.0
      IF(YYY.LE.20000.0)GOTO1035
1036 FORMAT(1H0,5X,'T TEST FAILED IN ABSORB')
      CONTINUE
1035 IF(IGRAY.EQ.1)GO TO 1037
      GO TO (1037,1038,1039,1040,1041),ISTEP
1037 IF(YYY.LT.8000.0)GO TO 1042
      IF(YYY.LE.10000.0)GOTO1043
      IF(YYY.LE.12000.0)GOTO1044
      IF(YYY.LE.14000.0)GOTO1045
      IF(YYY.LE.16000.0)GOTO1046
      IF(YYY.LE.18000.0)GOTO1047
      GOTO1241
1042 ABSORB=0.0
      GO TO 1048
1043 ABSORB=(1.0061D-13)*(10.0**(-2.6064*YY))
      GO TO 1049
1044 ABSORB=(9.6882D-15)*(10.0**(-1.5953*YY))
      GO TO 1049
1045 ABSORB=(2.0340D-15)*(10.0**(-1.0304*YY))
      GO TO 1049
1046 ABSORB=(6.9685D-16)*(10.0**(-0.6981*YY))
      GO TO 1049
1047 ABSORB=(3.3188D-16)*(10.0**(-0.49675*YY))
      GOTO1049
1241 ABSORB=(2.0976D-16)*(10.0**(-0.38605*YY))
1049 FNDEN=RCON(K,3,I)
      EDEKT=(4.707D-06)*(RCON(K,7,I)**(2.0/7.0))/(((0.86176D-04)*YYY)**
11.5)
      EDEKT=DEXP(EDEKT)
      ABSORB=(ABSORB*EDEKT)*FNDEN
1048 ABSOR1=ABSORB
      IF(IGRAY.EQ.1)GOTO1038
      RETURN
1038 IF(YYY.LT.8000.0)GOTO1050
      IF(YYY.LE.10000.0)GOTO1051
      IF(YYY.LE.12000.0)GOTO1052
      IF(YYY.LE.14000.0)GOTO1053
      IF(YYY.LE.16000.0)GOTO1054
      IF(YYY.LE.18000.0)GOTO1055
      GOTO1242
1050 ABSORB=0.0
      GOTO1056
1051 ABSORB=(5.9872D-22)*(10.0**(2.01935*YY))
      GOTO1057
1052 ABSORB=(3.2660D-21)*(10.0**(1.28255*YY))
      GOTO1057
1053 ABSORB=(1.1243D-20)*(10.0**(0.83515*YY))
      GOTO1057
1054 ABSORB=(2.1562D-20)*(10.0**(0.63315*YY))
      GOTO1057
1055 ABSORB=(3.7012D-20)*(10.0**(0.4865*YY))

```

```

GOTO1057
1242 ABSORB=(5.4757D-20)*(10.0**(0.3924*YY))
1057 FNDEN=RCON(K,3,I)
EDEKT=(4.707D-06)*(RCON(K,7,I)**(2.0/7.0))/(((0.86176D-04)*YYY)**
11.5)
EDEKT=DEXP(EDEKT)
ABSORB=(ABSORB*EDEKT)*FNDEN
1056 ABSOR2=ABSORB
IF(IGRAY.EQ.1)GOTO1039
RETURN
1039 IF(YYY.LT.8000.0)GOTO1058
IF(YYY.LE.10000.0)GOTO1059
IF(YYY.LE.12000.0)GOTO1060
IF(YYY.LE.14000.0)GOTO1061
IF(YYY.LE.16000.0)GOTO1062
IF(YYY.LE.18000.0)GOTO1063
GOTO1243
1058 ABSORB=0.0
GOTO1064
1059 ABSORB=((7.266D-20)*(10.0**(21.267*YY)))*(1.0D-20)
GOTO1065
1060 ABSORB=(2.2772D-19)*(10.0**(0.77045*YY))
GOTO1065
1061 ABSORB=(4.8471D-19)*(10.0**(0.49705*YY))
GOTO1065
1062 ABSORB=(8.1764D-19)*(10.0**(0.33485*YY))
GOTO1065
1063 ABSORB=(1.1902D-18)*(10.0**(0.23295*YY))
GOTO1065
1243 ABSORB=(2.19875D-18)*(10.0**(0.08485*YY))
1065 FNDEN=RCON(K,3,I)
ABSORB=ABSORB*FNDEN
1064 ABSOR3=ABSORB
IF(IGRAY.EQ.1)GOTO1040
RETURN
1040 IF(YYY.LT.8000.0)GOTO1066
IF(YYY.LE.10000.0)GOTO1067
IF(YYY.LE.12000.0)GOTO1068
IF(YYY.LE.14000.0)GOTO1069
IF(YYY.LE.16000.0)GOTO1070
IF(YYY.LE.18000.0)GOTO1071
GOTO1244
1066 ABSORB=0.0
GOTO1072
1067 ABSORB=(3.4065D-30)*(10.0**(7.2773*YY))
GOTO1073
1068 ABSORB=(8.3188D-28)*(10.0**(4.8895*YY))
GOTO1073
1069 ABSORB=(2.8643D-26)*(10.0**(3.6087*YY))
GOTO1073
1070 ABSORB=(1.9928D-25)*(10.0**(3.00695*YY))
GOTO1073
1071 ABSORB=(3.9376D-24)*(10.0**(2.1971*YY))
GOTO1073
1244 ABSORB=(4.2515D-24)*(10.0**(2.1786*YY))
1073 FNDEN=RCON(K,3,I)
EDEKT=(4.707D-06)*(RCON(K,7,I)**(2.0/7.0))/(((0.86176D-04)*YYY)**
11.5)
EDEKT=DEXP(EDEKT)
ABSORB=(ABSORB*EDEKT)*FNDEN
1072 ABSOR4=ABSORB
IF(IGRAY.EQ.1)GOTO1041
RETURN
1041 IF(YYY.LT.8000.0)GOTO1074
IF(YYY.LE.10000.0)GOTO1075
IF(YYY.LE.12000.0)GOTO1076
IF(YYY.LE.14000.0)GOTO1077
IF(YYY.LE.16000.0)GOTO1078
IF(YYY.LE.18000.0)GOTO1079
GOTO1245
1074 ABSORB=0.0
GOTO1080
1075 ABSORB=(3.1595D-27)*(10.0**(6.54335*YY))
GOTO1081
1076 ABSORB=(6.8837D-25)*(10.0**(4.20515*YY))
GOTO1081
1077 ABSORB=(1.8079D-23)*(10.0**(3.02235*YY))
GOTO1081
1078 ABSORB=(4.1675D-22)*(10.0**(2.04900*YY))
GOTO1081

```

```

1079 ABSORB=(2.4630D-21)*(10.0**(1.56667*YY))
      GOTO1081
1245 ABSORB=(1.0535D-20)*(10.0**(1.2161*YY))
1081 FNDEN=RCON(K,3,1)
      ABSORB=ABSORB*FNDEN
1080 ABSOR5=ABSORB
      IF(IGRAY.EQ.1)GOTO1082
      RETURN
1082 ABSORB=(ABSOR1*B(K,1,1)+ABSOR2*B(K,2,1)+ABSOR3*B(K,3,1)+
1ABSOR4*B(K,4,1)+ABSOR5*B(K,5,1))/BLACKB
      ABSORB=ABSORB*(1.0D+07)
      RETURN
      END

```

C
C-----
C

```

SUBROUTINE CORRAD (IQ4,IQ5,IQ6,IQ8,IQ9)
IMPLICIT REAL*8(A-H,O-Z)
COMMON /A1/ TE1,TE2,M,IX,IMAX,IPSI,MODEL
COMMON /A3/ EVIINF(25),THETAI(25),MUI(25),FI(25),DELHI(25),
* CIINF(25),LI(25)
COMMON /A5/ EPSIIL(20,25),GIL(20,25),CID(200,50)
COMMON /A9/ EVIS(25),XPST(100)
COMMON /RAD/ RCON(100,10,40),RRHO(100,40),TEMP(100,40)
* ,TSTAG,Y(100,40),NXCON,NPTS(100)
COMMON /RADC/ YBDY(100),B(100,8,40),DPTL(100,8,40)
COMMON /RADD/ TEMPD(100,40),ALPN(100,40),ALPD(100,40)
COMMON /RADDL/ BETN(100,40),BETO(100,40)
DIMENSION CR(40,13)
REAL*8 KAPPA(100,8,40),MUI
WRITE(8,1)
1 FORMAT(///' CORNELL RADIATION MODEL')
IF(IQ8.EQ.0) THEN
  IF(IQ6.EQ.0) WRITE(8,2)
  IF(IQ6.EQ.1) WRITE(8,3)
ENDIF
2 FORMAT('/' BETA=BETA(Te)')
3 FORMAT('/' BETA=BETA(Tt)')

```

C
C

```

NUMBER DENSITIES
AND NONEQUILIBRIUM CORRECTION
IF((IQ4.EQ.5).OR.(IQ4.EQ.6)) GO TO 1111
DO 10 K=3,NXCON
DO 10 J=1,NPTS(K)
DO 10 II=1,IMAX
XXX=RRHO(K,J)*6.023D+23
XXX=XXX*RCON(K,II,J)/MUI(II)
10 RCON(K,II,J)=XXX
IF (IQ5.EQ.0) THEN
  DO 122 K=3,NXCON
  DO 122 J=1,NPTS(K)
  IF(J.EQ.NPTS(K)) THEN
    ALPN(K,J)=0.D+00
    BETN(K,J)=0.D+00
    GO TO 122
  ENDIF
  Q1=0.D+0
  Q2=0.D+0
  Q3=0.D+0
  DO 112 III=1,LI(3)
112 Q1=Q1+GIL(III,3)*DEXP(-EPSIIL(III,3)/TEMP(K,J))
  DO 113 III=1,LI(9)
113 Q2=Q2+GIL(III,9)*DEXP(-EPSIIL(III,9)/TEMP(K,J))
  DO 130 III=1,LI(1)
130 Q3=Q3+GIL(III,1)*DEXP(-EPSIIL(III,1)/TEMP(K,J))
  RHO1=RCON(K,3,J)*1.401D+01/6.023D+23
  RHO2=RCON(K,9,J)*1.401D+01/6.023D+23
  RHO3=RCON(K,1,J)*2.802D+01/6.023D+23
  RHO4=RHO1+RHO2+RHO3
  QVIB=1.D+0/(1.D+0-DEXP(-3.39D+03/TEMP(K,J)))
  QROT=TEMP(K,J)/5.8D+0
  C1=RHO4*2.4701D+01*QVIB*QROT*Q3*DEXP(1.135D+05/TEMP(K,J))
  C1=C1/(Q1**2*TEMP(K,J)**1.5)
  C2=RHO4*8.909D+06*Q1*DEXP(1.69D+05/TEMP(K,J))
  C2=C2/(Q2*TEMP(K,J)**1.5)
  ALPHA=0.5D+0
  BETA=0.5D+0
132 F=BETA**2*(1.D+0-ALPHA)**2*C1+BETA-1.D+0
  FA=-2.D+0*BETA**2*(1.D+0-ALPHA)*C1
  FB=2.D+0*BETA*(1.D+0-ALPHA)**2*C1+1.D+0
  G=ALPHA**2*BETA*C2+ALPHA-1.D+0

```

```

GA=2.D+O*ALPHA*BETA*C2+1.D+O
GB=ALPHA**2*C2
DENOM=FA*GB-FB*GA
DA=(-F*GB+G*FB)/DENOM
DB=(-G*FA+F*GA)/DENOM
ALPHA=ALPHA+DA
BETA=BETA+DB
IF((DABS(DA).GT.1.D-O5).OR.(DABS(DB).GT.1.D-5)) GO TO 132
ALPH=RCON(K,9,J)/(RCON(K,9,J)+RCON(K,3,J))
BET=(RCON(K,9,J)+RCON(K,3,J))/(2.D+O*RCON(K,1,J)+
* RCON(K,9,J)+RCON(K,3,J))
IF(IQ6.EQ.1) GO TO 1888
BETN(K,J)=(BET**2/(1.D+O-BET))*((1.D+O-BETA)/(BETA**2))
* ((1.D+O-ALPH)**2/(1.D+O-ALPHA)**2)
1888 ALPN(K,J)=BET*(ALPH**2/(1.D+O-ALPH))
* ((1.D+O-ALPHA)/(BETA*ALPHA**2))
122 CONTINUE
DO 142 K=3,NXCON
DO 142 J=1,NPTS(K)
IF(J.EQ.NPTS(K)) THEN
ALPO(K,J)=O.D+OO
BETO(K,J)=O.D+OO
GO TO 142
ENDIF
Q1=O.D+O
Q2=O.D+O
Q3=O.D+O
DO 152 III=1,LI(4)
152 Q1=Q1+GIL(III,4)*EXP(-EPSIIL(III,4)/TEMP(K,J))
DO 153 III=1,LI(10)
153 Q2=Q2+GIL(III,10)*DEXP(-EPSIIL(III,10)/TEMP(K,J))
DO 160 III=1,LI(2)
160 Q3=Q3+GIL(III,2)*DEXP(-EPSIIL(III,2)/TEMP(K,J))
RH01=RCON(K,4,J)*1.6OOD+O1/6.O23D+23
RH02=RCON(K,10,J)*1.6OOD+O1/6.O23D+23
RH03=RCON(K,2,J)*3.2OOD+O1/6.O23D+23
RH04=RH01+RH02+RH03
QVIB=1.D+O/(1.D+O-DEXP(-2.27D+O3/TEMP(K,J)))
QROT=TEMP(K,J)/4.2D+O
C1=RH04*1.7725D+O1*QVIB*QROT*Q3*DEXP(5.95OD+O4/TEMP(K,J))
C1=C1/(Q1**2*TEMP(K,J)**1.5)
C2=RH04*7.8O1D+O6*Q1*DEXP(1.58D+O5/TEMP(K,J))
C2=C2/(Q2*TEMP(K,J)**1.5)
ALPHA=O.5D+O
BETA=O.5D+O
162 F=BETA**2*(1.D+O-ALPHA)**2*C1+BETA-1.D+O
FA=-2.D+O*BETA**2*(1.D+O-ALPHA)*C1
FB=2.D+O*BETA*(1.D+O-ALPHA)**2*C1+1.D+O
G=ALPHA**2*BETA*C2+ALPHA-1.D+O
GA=2.D+O*ALPHA*BETA*C2+1.D+O
GB=ALPHA**2*C2
DENOM=FA*GB-FB*GA
DA=(-F*GB+G*FB)/DENOM
DB=(-G*FA+F*GA)/DENOM
ALPHA=ALPHA+DA
BETA=BETA+DB
IF((DABS(DA).GT.1.D-O5).OR.(DABS(DB).GT.1.D-5)) GO TO 162
ALPH=RCON(K,10,J)/(RCON(K,10,J)+RCON(K,4,J))
BET=(RCON(K,10,J)+RCON(K,4,J))/(2.D+O*RCON(K,2,J)+
* RCON(K,10,J)+RCON(K,4,J))
IF(IQ6.EQ.1) GO TO 1999
BETO(K,J)=(BET**2/(1.D+O-BET))*((1.D+O-BETA)/(BETA**2))
* ((1.D+O-ALPH)**2/(1.D+O-ALPHA)**2)
1999 ALPO(K,J)=BET*(ALPH**2/(1.D+O-ALPH))
* ((1.D+O-ALPHA)/(BETA*ALPHA**2))
142 CONTINUE
C BETA=BETA(Tt)
IF(IQ6.EQ.1) THEN
DO 1122 K=3,NXCON
DO 1122 J=1,NPTS(K)
IF(J.EQ.NPTS(K)) THEN
BETN(K,J)=O.D+OO
GO TO 1122
ENDIF
Q1=O.D+O
Q2=O.D+O
Q3=O.D+O
DO 1112 III=1,LI(3)
1112 Q1=Q1+GIL(III,3)*DEXP(-EPSIIL(III,3)/TEMPO(K,J))
DO 1113 III=1,LI(9)

```

```

1113 Q2=Q2+GIL(III,9)*DEXP(-EPSIIL(III,9)/TEMPO(K,J))
DO 1130 III=1,LI(1)
1130 Q3=Q3+GIL(III,1)*DEXP(-EPSIIL(III,1)/TEMPO(K,J))
RHO1=RCON(K,3,J)*1.401D+01/6.023D+23
RHO2=RCON(K,9,J)*1.401D+01/6.023D+23
RHO3=RCON(K,1,J)*2.802D+01/6.023D+23
RHO4=RHO1+RHO2+RHO3
QVIB=1.D+O/(1.D+O-DEXP(-3.39D+03/TEMPO(K,J)))
QROT=TEMPO(K,J)/5.8D+O
C1=RHO4*2.4701D+01*QVIB*QROT*Q3*DEXP(1.135D+05/TEMPO(K,J))
C1=C1/(Q1**2*TEMPO(K,J)**1.5)
C2=RHO4*8.909D+06*Q1*DEXP(1.69D+05/TEMPO(K,J))
C2=C2/(Q2*TEMPO(K,J)**1.5)
ALPHA=O.5D+O
BETA=O.5D+O
1132 F=BETA**2*(1.D+O-ALPHA)**2*C1+BETA-1.D+O
FA=-2.D+O*BETA**2*(1.D+O-ALPHA)*C1
FB=2.D+O*BETA*(1.D+O-ALPHA)**2*C1+1.D+O
G=ALPHA**2*BETA*C2+ALPHA-1.D+O
GA=2.D+O*ALPHA*BETA*C2+1.D+O
GB=ALPHA**2*C2
DENOM=FA*GB-FB*GA
DA=(-F*GB+G*FB)/DENOM
DB=(-G*FA+F*GA)/DENOM
ALPHA=ALPHA+DA
BETA=BETA+DB
IF((DABS(DA).GT.1.D-05).OR.(DABS(DB).GT.1.D-5)) GO TO 1132
BET=(RCON(K,9,J)+RCON(K,3,J))/(2.D+O*RCON(K,1,J)+
* RCON(K,9,J)+RCON(K,3,J))
ALPH=RCON(K,9,J)/(RCON(K,9,J)+RCON(K,3,J))
BETN(K,J)=(BET**2/(1.D+O-BET))*((1.D+O-BETA)/(BETA**2))
* ((1.D+O-ALPH)**2/(1.D+O-ALPHA)**2)
1122 CONTINUE
DO 1142 K=3,NXCON
DO 1142 J=1,NPTS(K)
IF(J.EQ.NPTS(K)) THEN
BETO(K,J)=O.D+OO
GO TO 1142
ENDIF
Q1=O.D+O
Q2=O.D+O
Q3=O.D+O
DO 1152 III=1,LI(4)
1152 Q1=Q1+GIL(III,4)*EXP(-EPSIIL(III,4)/TEMPO(K,J))
DO 1153 III=1,LI(10)
1153 Q2=Q2+GIL(III,10)*DEXP(-EPSIIL(III,10)/TEMPO(K,J))
DO 1160 III=1,LI(2)
1160 Q3=Q3+GIL(III,2)*DEXP(-EPSIIL(III,2)/TEMPO(K,J))
RHO1=RCON(K,4,J)*1.600D+01/6.023D+23
RHO2=RCON(K,10,J)*1.600D+01/6.023D+23
RHO3=RCON(K,2,J)*3.200D+01/6.023D+23
RHO4=RHO1+RHO2+RHO3
QVIB=1.D+O/(1.D+O-DEXP(-2.27D+03/TEMPO(K,J)))
QROT=TEMPO(K,J)/4.2D+O
C1=RHO4*1.7725D+01*QVIB*QROT*Q3*DEXP(5.950D+04/TEMPO(K,J))
C1=C1/(Q1**2*TEMPO(K,J)**1.5)
C2=RHO4*7.801D+06*Q1*DEXP(1.58D+05/TEMPO(K,J))
C2=C2/(Q2*TEMPO(K,J)**1.5)
ALPHA=O.5D+O
BETA=O.5D+O
1162 F=BETA**2*(1.D+O-ALPHA)**2*C1+BETA-1.D+O
FA=-2.D+O*BETA**2*(1.D+O-ALPHA)*C1
FB=2.D+O*BETA*(1.D+O-ALPHA)**2*C1+1.D+O
G=ALPHA**2*BETA*C2+ALPHA-1.D+O
GA=2.D+O*ALPHA*BETA*C2+1.D+O
GB=ALPHA**2*C2
DENOM=FA*GB-FB*GA
DA=(-F*GB+G*FB)/DENOM
DB=(-G*FA+F*GA)/DENOM
ALPHA=ALPHA+DA
BETA=BETA+DB
IF((DABS(DA).GT.1.D-05).OR.(DABS(DB).GT.1.D-5)) GO TO 1162
BET=(RCON(K,10,J)+RCON(K,4,J))/(2.D+O*RCON(K,2,J)+
* RCON(K,10,J)+RCON(K,4,J))
ALPH=RCON(K,10,J)/(RCON(K,10,J)+RCON(K,4,J))
BETO(K,J)=(BET**2/(1.D+O-BET))*((1.D+O-BETA)/(BETA**2))
* ((1.D+O-ALPH)**2/(1.D+O-ALPHA)**2)
1142 CONTINUE
ENDIF
ENDIF

```

1111 CONTINUE

C

QRW

```

DO 200 K=3,NXCON
WRITE(8,102) XPST(K)
102 FORMAT(///,' AT X = ',D11.4)
WQR=0.0
DO 120 J=1,NPTS(K)
TT=TEMP(K,J)
CR(J,1)=3.6D-13*RCON(K,2,1)*DEXP(-5.7D+4/TT)
CR(J,2)=1.8D-14*RCON(K,1,1)*DEXP(-8.3D+4/TT)
CR(J,3)=3.0D-13*RCON(K,5,1)*DEXP(-6.7D+4/TT)
CR(J,4)=2.1D-19*RCON(K,5,1)*DEXP(-7.6D+3/TT)
SM=2.0D-20*(RCON(K,3,1)+RCON(K,7,1))*DEXP(-5.2D+04/TT)
CR(J,5)=SM*1.0D-14
SM=3.0D-20*(RCON(K,4,1)+RCON(K,7,1))*DEXP(-4.2D+04/TT)
CR(J,6)=SM*1.0D-14
IF(IQ5.EQ.0) THEN
CR(J,7)=1.0D-10*RCON(K,3,J)*DEXP(-1.8D+05/TT)*ALPN(K,J)
CR(J,8)=8.0D-13*RCON(K,3,J)*DEXP(-5.9D+04/TT)*ALPN(K,J)
CR(J,9)=9.0D-11*RCON(K,4,J)*DEXP(-1.7D+05/TT)*ALPD(K,J)
CR(J,10)=8.0D-13*RCON(K,4,J)*DEXP(-5.9D+04/TT)*ALPD(K,J)
IF(IQ8.EQ.0) THEN
CR(J,13)=1.6D-12*RCON(K,8,J)*DEXP(-3.6D+04/TT)*BETN(K,J)
ELSE
CR(J,13)=1.6D-12*RCON(K,8,J)*DEXP(-3.6D+04/TT)
ENDIF
ELSE
CR(J,7)=1.0D-10*RCON(K,3,J)*DEXP(-1.8D+05/TT)
CR(J,8)=8.0D-13*RCON(K,3,J)*DEXP(-5.9D+04/TT)
CR(J,9)=9.0D-11*RCON(K,4,J)*DEXP(-1.7D+05/TT)
CR(J,10)=8.0D-13*RCON(K,4,J)*DEXP(-5.9D+04/TT)
CR(J,13)=1.6D-12*RCON(K,8,J)*DEXP(-3.6D+04/TT)
ENDIF
CR(J,11)=9.0D-33*(RCON(K,9,J)+RCON(K,7,J))
**DEXP(-1.4D+04/TT)
CR(J,12)=9.0D-33*(RCON(K,10,J)+RCON(K,7,J))
**DEXP(-1.4D+04/TT)
120 CONTINUE
DO 180 I=1,13
JIN=1
RSUM=3.1415927D+00*CR(1,I)*DABS(Y(K,1)-YBDY(K))
IF(IQ9.EQ.0) WRITE(8,2221) JIN,I,RSUM
DO 150 J=2,NPTS(K)
STSUM=0.5D+00*(CR(J,I)+CR(J-1,I))*DABS(Y(K,J-1)-Y(K,J))
C *2.0D+00*3.1415927
IF(IQ9.EQ.0) THEN
WRITE(8,2221) J,I,STSUM
2221 FORMAT(1X,' AT J = ',I2,' EQUATION ',I2,' QR = ',D11.4)
ENDIF
RSUM=RSUM+STSUM
150 CONTINUE
WRITE(8,103) I,RSUM
103 FORMAT(/,' FOR EQUATION ',I2,' QR = ',D11.4,' WATTS/SQ.CM.')
WQR=WQR+RSUM
180 CONTINUE
WRITE(8,101) WQR
101 FORMAT(/' TOTAL QR = ',D11.4,' WATTS/SQ.CM.')
200 CONTINUE
RETURN
END

```

C

C

```

SUBROUTINE ANDRAD (IQ4,IQ5,IQ9)
IMPLICIT REAL*8(A-H,O-Z)
COMMON /A1/ TE1,TE2,M,IX,IMAX,IPSI,MODEL
COMMON /A3/ EVIINF(25),THETAI(25),MUI(25),FI(25),DELHI(25),
* CIINF(25),LI(25)
COMMON /A5/ EPSIIL(20,25),GIL(20,25),CID(200,50)
COMMON /A9/ EVIS(25),XPST(100)
COMMON /RAD/ RCON(100,10,40),RRHO(100,40),TEMP(100,40)
* ,TSTAG,Y(100,40),NXCON,NPTS(100)
COMMON /RADC/ YBDY(100),B(100,8,40),OPTL(100,8,40)
COMMON /RADD/ TEMPO(100,40),ALPN(100,40),ALPD(100,40)
REAL*8 KAPPA(100,8,40),MUI
WRITE(8,1)
1 FORMAT(///' ANDERSON RADIATION MODEL')
C
C NUMBER DENSITIES
IF((IQ4.EQ.5).OR.(IQ4.EQ.6)) GO TO 1111
DO 10 K=3,NXCON

```

```

DO 10 J=1,NPTS(K)
DO 10 II=1,IMAX
XXX=RRHO(K,J)*6.023D+23
XXX=XXX*RCON(K,II,J)/MUI(II)
10 RCON(K,II,J)=XXX
C      NONEQUILIBRIUM RADIATION CORRECTION
IF (IQ5.EQ.0) THEN
DO 122 K=3,NXCON
DO 122 J=1,NPTS(K)
IF(J.EQ.NPTS(K)) THEN
ALPN(K,J)=0.D+00
GO TO 122
ENDIF
Q1=0.D+0
Q2=0.D+0
Q3=0.D+0
DO 112 III=1,LI(3)
112 Q1=Q1+GIL(III,3)*DEXP(-EPSIIL(III,3)/TEMP(K,J))
DO 113 III=1,LI(9)
113 Q2=Q2+GIL(III,9)*DEXP(-EPSIIL(III,9)/TEMP(K,J))
DO 130 III=1,LI(1)
130 Q3=Q3+GIL(III,1)*DEXP(-EPSIIL(III,1)/TEMP(K,J))
RH01=RCON(K,3,J)*1.401D+01/6.023D+23
RH02=RCON(K,9,J)*1.401D+01/6.023D+23
RH03=RCON(K,1,J)*2.802D+01/6.023D+23
RH04=RH01+RH02+RH03
QVIB=1.D+0/((1.D+0-DEXP(-3.39D+03/TEMP(K,J)))
QROT=TEMP(K,J)/5.8D+0
C1=RH04*2.4701D+01*QVIB*QROT*Q3*DEXP(1.135D+05/TEMP(K,J))
C1=C1/(Q1**2*TEMP(K,J)**1.5)
C2=RH04*8.909D+06*Q1*DEXP(1.69D+05/TEMP(K,J))
C2=C2/(Q2*TEMP(K,J)**1.5)
ALPHA=0.5D+0
BETA=0.5D+0
132 F=BETA**2*(1.D+0-ALPHA)**2*C1+BETA-1.D+0
FA=-2.D+0*BETA**2*(1.D+0-ALPHA)*C1
FB=2.D+0*BETA*(1.D+0-ALPHA)**2*C1+1.D+0
G=ALPHA**2*BETA*C2+ALPHA-1.D+0
GA=2.D+0*ALPHA*BETA*C2+1.D+0
GB=ALPHA**2*C2
DENOM=FA*GB-FB*GA
DA=(-F*GB+G*FB)/DENOM
DB=(-G*FA+F*GA)/DENOM
ALPHA=ALPHA+DA
BETA=BETA+DB
IF((DABS(DA).GT.1.D-05).OR.(DABS(DB).GT.1.D-5)) GO TO 132
ALPH=RCON(K,9,J)/(RCON(K,9,J)+RCON(K,3,J))
BET=(RCON(K,9,J)+RCON(K,3,J))/(RCON(K,9,J)+RCON(K,3,J)
+2.D+0*RCON(K,1,J))
ALPN(K,J)=(ALPH**2/((1.D+0-ALPH))*((1.D+0-ALPHA)/(ALPHA**2))
* (BET/BETA)
122 CONTINUE
ENDIF
1111 CONTINUE
C      DENSITY RATIOS
DO 12 K=3,NXCON
DO 12 J=1,NPTS(K)
IF(IQ4.EQ.4) RCON(K,7,J)=RCON(K,7,J)*RRHO(K,J)*6.023D+23/MUI(7)
RRHO(K,J)=RRHO(K,J)/1.2250D-03
12 CONTINUE
C      PLANCK FUNCTION
DO 30 K=3,NXCON
DO 30 J=1,NPTS(K)
TT=TEMP(K,J)/168800.0
IF(TEMP(K,J).LE.8.D+03) GO TO 30
BLACK=(TT*168800.0)**4*5.6696D-12/3.1415927
FF=15.D+00/(3.1415927**4)
X1=1./TT
B1=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C DEXP(-2.*X1)*0.5*(.75+1.5*X1+1.5*X1**2+X1**3))
X1=0.935/TT
B2=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C DEXP(-2.*X1)*0.5*(.75+1.5*X1+1.5*X1**2+X1**3))
X1=0.835/TT
B3=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C DEXP(-2.*X1)*0.5*(.75+1.5*X1+1.5*X1**2+X1**3))
X1=0.754/TT
B4=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C DEXP(-2.*X1)*0.5*(.75+1.5*X1+1.5*X1**2+X1**3))
X1=0.473/TT

```



```

B6=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C   DEXP(-2.*X1)*O.5*(.75+1.5*X1+1.5*X1**2+X1**3))
X1=O.213/TT
B8=FF*(DEXP(-X1)*(6.+6.*X1+3.*X1**2+X1**3)+
C   DEXP(-2.*X1)*O.5*(.75+1.5*X1+1.5*X1**2+X1**3))
B8=1-B8
B2=B2-B1
B3=B3-B2
B4=B4-B3
B6=B6-B4
B5=1-B8-B6-B4-B3-B2-B1
B(K,1,J)=BLACK*B1
B(K,2,J)=BLACK*B2
B(K,3,J)=BLACK*B3
B(K,4,J)=BLACK*B4
IF(IQ5.EQ.O) THEN
  B(K,1,J)=B(K,1,J)*ALPN(K,J)
  B(K,2,J)=B(K,2,J)*ALPN(K,J)
  B(K,3,J)=B(K,3,J)*ALPN(K,J)
  B(K,4,J)=B(K,4,J)*ALPN(K,J)
ENDIF
B(K,6,J)=BLACK*B6
B(K,7,J)=BLACK*2.4D-21*RCON(K,7,J)*B6*DEXP(.162/TT)
B(K,8,J)=BLACK*B8
B(K,5,J)=BLACK*B5
B(K,1,J)=B(K,1,J)+B(K,2,J)+B(K,3,J)+B(K,4,J)
B(K,2,J)=B(K,5,J)+B(K,6,J)+B(K,7,J)+B(K,8,J)
30 CONTINUE
C
ABSORPTION COEFFICIENTS
DO 20 K=3,NXCON
DO 20 J=1,NPTS(K)
TT=TEMP(K,J)/1.0D+04
IF(TEMP(K,J).LE.1.1D+04) THEN
  KAPPA(K,1,J)=36.D+00*RRHO(K,J)*TT**4.O2
ELSE
  KAPPA(K,1,J)=RRHO(K,J)*(8.1D+00+41.3D+00*TT)
ENDIF
IF(TEMP(K,J).LE.7.D+03) THEN
  AA=234.7D+00
  BB=1.41D+00
  CC=7.83D+00
  GO TO 25
ENDIF
IF(TEMP(K,J).LE.9.D+03) THEN
  AA=262.D+00
  BB=1.4D+00
  CC=8.19D+00
  GO TO 25
ENDIF
IF(TEMP(K,J).LE.1.1D+04) THEN
  AA=266.D+00
  BB=1.15D+00
  CC=13.86D+00
  GO TO 25
ENDIF
IF(TEMP(K,J).LE.1.3D+04) THEN
  AA=232.D+00
  BB=1.06D+00
  CC=13.1D+00
  GO TO 25
ENDIF
AA=406.5D+00
BB=1.1D+00
CC=11.3D+00
25 JP=AA*RRHO(K,J)**BB*TT**CC
IF(B(K,2,J).LE.1.D-05) THEN
  KAPPA(K,2,J)=O.D+00
  GO TO 20
ENDIF
KAPPA(K,2,J)=JP/B(K,2,J)
IF(IQ5.EQ.O) THEN
  KAPPA(K,2,J)=KAPPA(K,2,J)*ALPN(K,J)
ENDIF
20 CONTINUE
C
TAU'S
DO 100 K=3,NXCON
DO 100 I=1,2
OPTL(K,I,1)=5.D-01*KAPPA(K,I,1)*DABS(Y(K,1)-YBDY(K))
DO 100 J=2,NPTS(K)
OPTL(K,I,J)=OPTL(K,I,J-1)+(O.5*(KAPPA(K,I,J)+KAPPA(K,I,J-1)))*

```

```

      C      DABS(Y(K,J-1)-Y(K,J)))
100 CONTINUE
C      QRW
      DO 200 K=3,NXCON
      WRITE(8,102) XPST(K)
102  FORMAT(///,' AT X = ',D11.4)
      WQR=0.0
      DO 180 I=1,2
      CALL FEI(E1I,E2I,E3I,OPTL(K,I,1))
      EOLD=E2I
      JIN=1
      SUM=3.1415927*EOLD*B(K,I,1)*KAPPA(K,I,1)*DABS(Y(K,1)-YBDY(K))
      IF(IQ9.EQ.0) WRITE(8,2221) JIN,I,SUM
      DO 150 J=2,NPTS(K)
      Z=OPTL(K,I,J)
      IF(Z.EQ.0.D+0) THEN
      E2I=1.D+0
      GO TO 145
      ENDIF
      CALL FEI(E1I,E2I,E3I,Z)
      STSUM=3.14159*(B(K,I,J)*KAPPA(K,I,J)*E2I+B(K,I,J-1)*
C      KAPPA(K,I,J-1)*EOLD)*DABS(Y(K,J-1)-Y(K,J))
      IF(IQ9.EQ.0) THEN
      WRITE(8,2221) J,I,STSUM
2221  FORMAT(1X,' AT J = ',I2,' BAND ',I2,' QR = ',D11.4)
      ENDIF
145  SUM=SUM+STSUM
      EOLD=E2I
150  CONTINUE
      WRITE(8,103) I,SUM
103  FORMAT(/,' FOR BAND ',I2,' QR = ',D11.4,' WATTS/SQ.CM.')
      WQR=WQR+SUM
180  CONTINUE
      WRITE(8,101) WQR
101  FORMAT(/' TOTAL QR = ',D11.4,' WATTS/SQ.CM.')
200  CONTINUE
      IF((IQ4.EQ.5).OR.(IQ4.EQ.6)) THEN
      DO 1000 K=3,NXCON
      DO 1000 J=1,NPTS(K)
      RRHO(K,J)=RRHO(K,J)*1.225D-03
1000  CONTINUE
      ENDIF
      RETURN
      END

```

LISTING OF INPUT FILE KDRW

0,0,1,0,0,0,1,0,1,0,0
 0,0,1,0,0,0,0,0,1,0,0
 0,0,0,0,0,0,1,0,2,0,0
 0,0,1,0,0,0,0,1,0,0,0
 0,0,0,0,0,0,2,0,1,0,0
 0,0,0,1,0,0,1,0,0,0,0
 0,0,0,0,0,0,2,0,0,1,0
 0,0,0,2,0,0,0,0,0,0,0
 0,1,0,0,0,0,0,0,0,0,0
 0,0,1,1,0,0,0,0,0,0,0
 0,0,0,0,1,0,0,0,0,0,0
 0,0,2,0,0,0,0,0,0,0,0
 1,0,0,0,0,0,0,0,0,0,0
 0,0,0,1,1,0,0,0,0,0,0
 0,1,1,0,0,0,0,0,0,0,0
 0,0,1,0,1,0,0,0,0,0,0
 1,0,0,1,0,0,0,0,0,0,0
 0,0,1,1,0,0,0,0,0,0,0
 0,0,0,0,0,1,1,0,0,0,0
 0,0,0,0,0,0,1,1,0,0,0
 0,0,2,0,0,0,0,0,0,0,0
 0,0,1,0,0,0,1,0,1,0,0
 0,0,2,0,0,0,0,0,0,0,0
 0,0,0,0,0,0,1,0,2,0,0
 0,0,1,0,0,0,0,0,1,0,0
 0,0,0,0,0,0,2,0,1,0,0
 0,0,1,0,0,0,0,1,0,0,0
 0,0,0,0,0,0,2,0,0,1,0
 0,0,0,1,0,0,1,0,0,0,0
 5.675E-04,5.675E-04,5.675E-04,5.675E-04,5.675E-04,5.675E-04,5.675E-04,
 5.675E-04,5.675E-04
 1.443E-07,1.443E-07,1.443E-07,1.443E-07,1.443E-07,1.443E-07,1.443E-07,1.443E-07,
 1.443E-07,1.443E-07
 0,0,0,0,0,0,0,0,0,0,0
 0,0,0,0,0,0,0,0,0,0,0
 1.513E-04,1.513E-04,1.513E-04,1.513E-04,1.513E-04,1.513E-04,1.513E-04,1.513E-04,
 1.513E-04,1.513E-04
 1.513E-04,1.513E-04,1.513E-04,1.513E-04,1.513E-04,1.513E-04,1.513E-04,1.513E-04,
 1.513E-04,1.513E-04
 0,0,0,0,0,0,0,0,0,0,0
 5.675E-04,5.675E-04,5.675E-04,5.675E-04,5.675E-04,5.675E-04,5.675E-04,5.675E-04,
 5.675E-04,5.675E-04
 0,0,0,0,0,0,0,0,0,0,0
 0,0,0,0,0,0,0,0,0,0,0
 -.09952,-.09952,-.09952,-.09952,-.09952,-.09952,-.09952,-.09952,-.09952
 .37177,.37177,.37177,.37177,.37177,.37177,.37177
 0,0,0,0,0,0,0,0,0,0,0
 0,0,0,0,0,0,0,0,0,0,0
 -.2226,-.2226,-.2226,-.2226,-.2226,-.2226,-.2226,-.2226,-.2226
 -.2226,-.2226,-.2226,-.2226,-.2226,-.2226,-.2226,-.2226,-.2226
 0,0,0,0,0,0,0,0,0,0,0
 -.09952,-.09952,-.09952,-.09952,-.09952,-.09952,-.09952,-.09952,-.09952
 0,0,0,0,0,0,0,0,0,0,0
 0,0,0,0,0,0,0,0,0,0,0
 163.63,163.63,163.63,163.63,163.63,163.63,163.63,163.63,163.63,163.63
 179.32,179.32,179.32,179.32,179.32,179.32,179.32,179.32,179.32,179.32
 0,0,0,0,0,0,0,0,0,0,0
 0,0,0,0,0,0,0,0,0,0,0
 135.3,135.3,135.3,135.3,135.3,135.3,135.3,135.3,135.3,135.3
 135.3,135.3,135.3,135.3,135.3,135.3,135.3,135.3,135.3,135.3
 0,0,0,0,0,0,0,0,0,0,0
 163.63,163.63,163.63,163.63,163.63,163.63,163.63,163.63,163.63,163.63
 0,0,0,0,0,0,0,0,0,0,0
 0,0,0,0,0,0,0,0,0,0,0
 1,3,6,2
 3,2,1,3,1,3
 4,6,4,6
 5,3,1,5,1
 2,2,2,4,2,4,2,4
 1,6,3
 1
 2,4,2,2
 1,3,5,5,1,5
 4,6,4,4,2
 0,71592,85343,99212
 0,11342,18879,51385,52104,71026
 0,27659,27670,41496
 0,228,326,22831,48622
 0,174,63597,65381,76676,75361,87568,86359
 0,57528,84205

O
 0,13191,36634,92977.4
 0,70.5,188.5,22036.5,47029.68,67864.8
 0,38571,38601,58223.5,58225
 46,18800,3393.9685,20.797195,1.0801997E-02,-7.32343E-04
 33,9896,2273.1279,17.365295,7.8529954E-02,-2.057468E-03
 0,0,0,0,0,0
 0,0,0,0,0,0
 0,0,0,0,0,0
 0,0,0,0,0,0
 0,0,0,0,0,0
 50,16850,3175.68,23.21629,-5.755154E-02,0
 0,0,0,0,0,0
 0,0,0,0,0,0
 N2
 O2
 N
 O
 NO
 NO+
 E-
 N2+
 N+
 O+
 O2 + M .> 2O + M
 2O + M .> O2 + M
 NO + M .> N + O + M
 N + O + M .> NO + M
 N2 + M .> 2N + M
 2N + M .> N2 + M
 N + O2 .> NO + O
 NO + O .> N + O2
 N2 + O .> NO + N
 NO + N .> N2 + O
 NO+ + E- .> N + O
 N + O .> NO+ + E-
 N + N .> N2+ + E-
 N2+ + E- .> N + N
 N + N .> N+ + E- + N
 N+ + E- + N .> N+ + N
 N + N+ .> N+ + E- + N+
 N+ + E- + N+ .> N+ + N+
 N + E- .> N+ + 2E-
 N+ + 2E- .> N+ + E-
 O + E- .> O+ + E- + E-
 O+ + E- + E- .> O+ + E-

PORTION OF A SAMPLE OUTPUT

SPECIES

```

1      N2
2      O2
3      N
4      O
5      NO
6      NO+
7      E-
8      N2+
9      N+
10     O+

```

REACTIONS

```

1      O2 + M -> 2O + M
2      2O + M -> O2 + M
3      NO + M -> N + O + M
4      N + O + M -> NO + M
5      N2 + M -> 2N + M
6      2N + M -> N2 + M
7      N + O2 -> NO + O
8      NO + O -> N + O2
9      N2 + O -> NO + N
10     NO + N -> N2 + O
11     NO + E- -> N + O
12     N + O -> NO + E-
13     N + N -> N2 + E-
14     N2 + E- -> N + N
15     N + N -> N+ + E- + N
16     N+ + E- + N -> N + N
17     N + N+ -> N+ + E- + N+
18     N+ + E- + N+ -> N + N+
19     N + E- -> N+ + 2E-
20     N+ + 2E- -> N + E-
21     O + E- -> O+ + E- + E-
22     O+ + E- + E- -> O + E-

```

HCHCKT = 0.1000E-01 TCHCKT = 0.3000E+00 PHMAX = 0.6500E+02

XI = 0.1221E-03 CIMAX = 0.5000E+00 NXPST = 20

ELE1(1) = 0.6400E+02	ELE2(1) = 0.1000E+01
ELE1(2) = 0.6400E+02	ELE2(2) = 0.1000E+01
ELE1(3) = 0.6400E+02	ELE2(3) = 0.1000E+01
ELE1(4) = 0.6400E+02	ELE2(4) = 0.1000E+01
ELE1(5) = 0.6400E+02	ELE2(5) = 0.1000E+01
ELE1(6) = 0.6400E+02	ELE2(6) = 0.1000E+01
ELE1(7) = 0.6400E+02	ELE2(7) = 0.1000E+01
ELE1(8) = 0.6400E+02	ELE2(8) = 0.1000E+01
ELE1(9) = 0.6400E+02	ELE2(9) = 0.1000E+01
ELE1(10) = 0.6400E+02	ELE2(10) = 0.1000E+01
ELE1(11) = 0.6400E+02	ELE2(11) = 0.1000E+01
ELE1(12) = 0.6400E+02	ELE2(12) = 0.1000E+01
ELE1(13) = 0.6400E+02	ELE2(13) = 0.1000E+01
ELE1(14) = 0.6400E+02	ELE2(14) = 0.1000E+01
ELE1(15) = 0.6400E+02	ELE2(15) = 0.1000E+01
ELE1(16) = 0.6400E+02	ELE2(16) = 0.1000E+01
ELE1(17) = 0.6400E+02	ELE2(17) = 0.1000E+01
ELE1(18) = 0.6400E+02	ELE2(18) = 0.1000E+01
ELE1(19) = 0.6400E+02	ELE2(19) = 0.1000E+01
ELE1(20) = 0.6400E+02	ELE2(20) = 0.1000E+01
ELE1(21) = 0.6400E+02	ELE2(21) = 0.1000E+01

XPST(1) = 0.5000E+00
XPST(2) = 0.1000E+01
XPST(3) = 0.1500E+01
XPST(4) = 0.2000E+01
XPST(5) = 0.2500E+01

XPST(6) = 0.3000E+01
 XPST(7) = 0.3500E+01
 XPST(8) = 0.4000E+01
 XPST(9) = 0.4500E+01
 XPST(10) = 0.5000E+01
 XPST(11) = 0.5500E+01
 XPST(12) = 0.6000E+01
 XPST(13) = 0.6500E+01
 XPST(14) = 0.7000E+01
 XPST(15) = 0.7500E+01
 XPST(16) = 0.8000E+01
 XPST(17) = 0.8500E+01
 XPST(18) = 0.9000E+01
 XPST(19) = 0.9500E+01
 XPST(20) = 0.1000E+02

IMAX = 10 JMAX = 22 M = 1
 DELX = 0.5000E+00 ZSTERM = 0.2000E+00 PRINT FREQ. = 50 R = 0.8315E+08
 PINF = 0.1572E+02 TINF = 0.1971E+03 VINI = 0.8915E+06 GAMMA = 0.140E+01

	MUI	THETAI	DGENI	FI	DELHI
1	0.2802E+02	0.3392E+04	0.1000E+01	0.1000E+01	0.0000E+00
2	0.3200E+02	0.2275E+04	0.1000E+01	0.1000E+01	0.0000E+00
3	0.1401E+02	0.1000E+01	0.1000E+01	0.0000E+00	0.4707E+13
4	0.1600E+02	0.1000E+01	0.1000E+01	0.0000E+00	0.2468E+13
5	0.3001E+02	0.2739E+04	0.1000E+01	0.1000E+01	0.9007E+12
6	0.3001E+02	0.3421E+04	0.1000E+01	0.1000E+01	0.1083E+14
7	0.5486E-03	0.1000E+01	0.1000E+01	0.0000E+00	0.0000E+00
8	0.2802E+02	0.3129E+04	0.1000E+01	0.1000E+01	0.1505E+14
9	0.1401E+02	0.1000E+01	0.1000E+01	0.0000E+00	0.1876E+14
10	0.1600E+02	0.1000E+01	0.1000E+01	0.0000E+00	0.1562E+14

	DELI	CIINF	EVI	LI	BI
1	0.1150E+06	0.7680E+00	0.0000E+00	4	-0.4216E+00
2	0.6050E+05	0.2320E+00	0.0000E+00	6	0.1216E+01
3	0.0000E+00	0.0000E+00	0.0000E+00	4	0.1680E+01
4	0.0000E+00	0.0000E+00	0.0000E+00	5	0.2103E+01
5	0.7688E+05	0.0000E+00	0.0000E+00	8	0.1233E+01
6	0.1277E+06	0.0000E+00	0.0000E+00	3	0.3786E+00
7	0.0000E+00	0.0000E+00	0.0000E+00	1	-0.1424E+02
8	0.1012E+06	0.0000E+00	0.0000E+00	4	0.1004E+01
9	0.0000E+00	0.0000E+00	0.0000E+00	6	0.2937E+00
10	0.0000E+00	0.0000E+00	0.0000E+00	5	0.1879E+01

	NIP	UP	WE	WEKE	WEYE	WEZE
1	46	0.1880E+05	0.3394E+04	0.2080E+02	0.1080E-01	-0.7323E-03
2	33	0.9896E+04	0.2273E+04	0.1737E+02	0.7853E-01	-0.2057E-02
3	0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
4	0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
5	0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
6	0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
7	0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
8	50	0.1685E+05	0.3176E+04	0.2322E+02	-0.5755E-01	0.0000E+00
9	0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
10	0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00

	MJ	AJ	BJ	EJ	DIRECTION F=1,B=2
1	2	0.119000E+22	-0.150000E+01	0.593800E+05	1.00
3	5	0.518000E+22	-0.150000E+01	0.754900E+05	1.00
5	3	0.500000E+20	-0.150000E+01	0.000000E+00	2.00
7	2	0.100000E+13	0.500000E+00	0.312000E+04	1.00
9	1	0.500000E+14	0.000000E+00	0.380160E+05	1.00
11	6	0.180000E+22	-0.150000E+01	0.000000E+00	1.00
13	3	0.140000E+14	0.000000E+00	0.678000E+05	1.00

15	3	0.234000E+12	0.500000E+00	0.120000E+06	1.00
17	9	0.234000E+12	0.500000E+00	0.120000E+06	1.00
19	3	0.416000E+14	0.500000E+00	0.120000E+06	1.00
21	4	0.549000E+14	0.500000E+00	0.104500E+06	1.00

	MJ	DIRECTION F=1,B=2
2	4	2.00
4	3	2.00
6	1	1.00
8	5	2.00
10	5	2.00
12	8	2.00
14	9	2.00
16	9	2.00
18	9	2.00
20	9	2.00
22	10	2.00

GIL(1, 1) = 0.1000E+01	EPSIIL(1, 1) = 0.0000E+00
GIL(2, 1) = 0.3000E+01	EPSIIL(2, 1) = 0.7159E+05
GIL(3, 1) = 0.6000E+01	EPSIIL(3, 1) = 0.8534E+05
GIL(4, 1) = 0.2000E+01	EPSIIL(4, 1) = 0.9921E+05
GIL(1, 2) = 0.3000E+01	EPSIIL(1, 2) = 0.0000E+00
GIL(2, 2) = 0.2000E+01	EPSIIL(2, 2) = 0.1134E+05
GIL(3, 2) = 0.1000E+01	EPSIIL(3, 2) = 0.1888E+05
GIL(4, 2) = 0.3000E+01	EPSIIL(4, 2) = 0.5139E+05
GIL(5, 2) = 0.1000E+01	EPSIIL(5, 2) = 0.5210E+05
GIL(6, 2) = 0.3000E+01	EPSIIL(6, 2) = 0.7103E+05
GIL(1, 3) = 0.4000E+01	EPSIIL(1, 3) = 0.0000E+00
GIL(2, 3) = 0.6000E+01	EPSIIL(2, 3) = 0.2766E+05
GIL(3, 3) = 0.4000E+01	EPSIIL(3, 3) = 0.2767E+05
GIL(4, 3) = 0.6000E+01	EPSIIL(4, 3) = 0.4150E+05
GIL(1, 4) = 0.5000E+01	EPSIIL(1, 4) = 0.0000E+00
GIL(2, 4) = 0.3000E+01	EPSIIL(2, 4) = 0.2280E+03
GIL(3, 4) = 0.1000E+01	EPSIIL(3, 4) = 0.3260E+03
GIL(4, 4) = 0.5000E+01	EPSIIL(4, 4) = 0.2283E+05
GIL(5, 4) = 0.1000E+01	EPSIIL(5, 4) = 0.4862E+05
GIL(1, 5) = 0.2000E+01	EPSIIL(1, 5) = 0.0000E+00
GIL(2, 5) = 0.2000E+01	EPSIIL(2, 5) = 0.1740E+03
GIL(3, 5) = 0.2000E+01	EPSIIL(3, 5) = 0.6360E+05
GIL(4, 5) = 0.4000E+01	EPSIIL(4, 5) = 0.6538E+05
GIL(5, 5) = 0.2000E+01	EPSIIL(5, 5) = 0.7668E+05
GIL(6, 5) = 0.4000E+01	EPSIIL(6, 5) = 0.7536E+05
GIL(7, 5) = 0.2000E+01	EPSIIL(7, 5) = 0.8757E+05
GIL(8, 5) = 0.4000E+01	EPSIIL(8, 5) = 0.8636E+05
GIL(1, 6) = 0.1000E+01	EPSIIL(1, 6) = 0.0000E+00
GIL(2, 6) = 0.6000E+01	EPSIIL(2, 6) = 0.5753E+05
GIL(3, 6) = 0.3000E+01	EPSIIL(3, 6) = 0.8421E+05
GIL(1, 7) = 0.1000E+01	EPSIIL(1, 7) = 0.0000E+00
GIL(1, 8) = 0.2000E+01	EPSIIL(1, 8) = 0.0000E+00
GIL(2, 8) = 0.4000E+01	EPSIIL(2, 8) = 0.1319E+05
GIL(3, 8) = 0.2000E+01	EPSIIL(3, 8) = 0.3663E+05
GIL(4, 8) = 0.2000E+01	EPSIIL(4, 8) = 0.9298E+05
GIL(1, 9) = 0.1000E+01	EPSIIL(1, 9) = 0.0000E+00
GIL(2, 9) = 0.3000E+01	EPSIIL(2, 9) = 0.7050E+02
GIL(3, 9) = 0.5000E+01	EPSIIL(3, 9) = 0.1885E+03
GIL(4, 9) = 0.5000E+01	EPSIIL(4, 9) = 0.2204E+05
GIL(5, 9) = 0.1000E+01	EPSIIL(5, 9) = 0.4703E+05
GIL(6, 9) = 0.5000E+01	EPSIIL(6, 9) = 0.6786E+05
GIL(1, 10) = 0.4000E+01	EPSIIL(1, 10) = 0.0000E+00
GIL(2, 10) = 0.6000E+01	EPSIIL(2, 10) = 0.3857E+05
GIL(3, 10) = 0.4000E+01	EPSIIL(3, 10) = 0.3860E+05
GIL(4, 10) = 0.4000E+01	EPSIIL(4, 10) = 0.5822E+05
GIL(5, 10) = 0.2000E+01	EPSIIL(5, 10) = 0.5823E+05

AIJ(1, 1) = 0.1000E+01
AIJ(1, 2) = 0.1000E+01
AIJ(1, 3) = 0.1000E+01
AIJ(1, 4) = 0.1000E+01
AIJ(1, 5) = 0.1000E+01
AIJ(1, 6) = 0.1000E+01
AIJ(1, 7) = 0.0000E+00

ORIGINAL PAGE IS
OF POOR QUALITY

AIJ(1, 8) = 0.0000E+00
 AIJ(1, 9) = 0.0000E+00
 AIJ(1,10) = 0.0000E+00
 AIJ(1,11) = 0.0000E+00
 AIJ(1,12) = 0.0000E+00
 AIJ(1,13) = 0.0000E+00
 AIJ(1,14) = 0.0000E+00
 AIJ(1,15) = 0.0000E+00
 AIJ(1,16) = 0.0000E+00
 AIJ(1,17) = 0.0000E+00
 AIJ(1,18) = 0.0000E+00
 AIJ(1,19) = 0.0000E+00
 AIJ(1,20) = 0.0000E+00
 AIJ(1,21) = 0.0000E+00
 AIJ(1,22) = 0.0000E+00
 AIJ(2, 1) = 0.1000E+01
 AIJ(2, 2) = 0.1000E+01
 AIJ(2, 3) = 0.1000E+01
 AIJ(2, 4) = 0.1000E+01
 AIJ(2, 5) = 0.1000E+01
 AIJ(2, 6) = 0.1000E+01
 AIJ(2, 7) = 0.0000E+00
 AIJ(2, 8) = 0.0000E+00
 AIJ(2, 9) = 0.0000E+00
 AIJ(2,10) = 0.0000E+00
 AIJ(2,11) = 0.0000E+00
 AIJ(2,12) = 0.0000E+00
 AIJ(2,13) = 0.0000E+00
 AIJ(2,14) = 0.0000E+00
 AIJ(2,15) = 0.0000E+00
 AIJ(2,16) = 0.0000E+00
 AIJ(2,17) = 0.0000E+00
 AIJ(2,18) = 0.0000E+00
 AIJ(2,19) = 0.0000E+00
 AIJ(2,20) = 0.0000E+00
 AIJ(2,21) = 0.0000E+00
 AIJ(2,22) = 0.0000E+00
 AIJ(3, 1) = 0.1000E+01
 AIJ(3, 2) = 0.1000E+01
 AIJ(3, 3) = 0.1000E+01
 AIJ(3, 4) = 0.1000E+01
 AIJ(3, 5) = 0.1000E+01
 AIJ(3, 6) = 0.1000E+01
 AIJ(3, 7) = 0.0000E+00
 AIJ(3, 8) = 0.0000E+00
 AIJ(3, 9) = 0.0000E+00
 AIJ(3,10) = 0.0000E+00
 AIJ(3,11) = 0.0000E+00
 AIJ(3,12) = 0.0000E+00
 AIJ(3,13) = 0.0000E+00
 AIJ(3,14) = 0.0000E+00
 AIJ(3,15) = 0.0000E+00
 AIJ(3,16) = 0.0000E+00
 AIJ(3,17) = 0.0000E+00
 AIJ(3,18) = 0.0000E+00
 AIJ(3,19) = 0.0000E+00
 AIJ(3,20) = 0.0000E+00
 AIJ(3,21) = 0.0000E+00
 AIJ(3,22) = 0.0000E+00
 AIJ(4, 1) = 0.1000E+01
 AIJ(4, 2) = 0.1000E+01
 AIJ(4, 3) = 0.1000E+01
 AIJ(4, 4) = 0.1000E+01
 AIJ(4, 5) = 0.1000E+01
 AIJ(4, 6) = 0.1000E+01
 AIJ(4, 7) = 0.0000E+00
 AIJ(4, 8) = 0.0000E+00
 AIJ(4, 9) = 0.0000E+00
 AIJ(4,10) = 0.0000E+00
 AIJ(4,11) = 0.0000E+00
 AIJ(4,12) = 0.0000E+00
 AIJ(4,13) = 0.0000E+00
 AIJ(4,14) = 0.0000E+00
 AIJ(4,15) = 0.0000E+00
 AIJ(4,16) = 0.0000E+00
 AIJ(4,17) = 0.0000E+00
 AIJ(4,18) = 0.0000E+00
 AIJ(4,19) = 0.0000E+00
 AIJ(4,20) = 0.0000E+00
 AIJ(4,21) = 0.0000E+00

AIJ(4,22) = 0.0000E+00
 AIJ(5, 1) = 0.1000E+01
 AIJ(5, 2) = 0.1000E+01
 AIJ(5, 3) = 0.1000E+01
 AIJ(5, 4) = 0.1000E+01
 AIJ(5, 5) = 0.1000E+01
 AIJ(5, 6) = 0.1000E+01
 AIJ(5, 7) = 0.0000E+00
 AIJ(5, 8) = 0.0000E+00
 AIJ(5, 9) = 0.0000E+00
 AIJ(5,10) = 0.0000E+00
 AIJ(5,11) = 0.0000E+00
 AIJ(5,12) = 0.0000E+00
 AIJ(5,13) = 0.0000E+00
 AIJ(5,14) = 0.0000E+00
 AIJ(5,15) = 0.0000E+00
 AIJ(5,16) = 0.0000E+00
 AIJ(5,17) = 0.0000E+00
 AIJ(5,18) = 0.0000E+00
 AIJ(5,19) = 0.0000E+00
 AIJ(5,20) = 0.0000E+00
 AIJ(5,21) = 0.0000E+00
 AIJ(5,22) = 0.0000E+00
 AIJ(6, 1) = 0.1000E+01
 AIJ(6, 2) = 0.1000E+01
 AIJ(6, 3) = 0.1000E+01
 AIJ(6, 4) = 0.1000E+01
 AIJ(6, 5) = 0.1000E+01
 AIJ(6, 6) = 0.1000E+01
 AIJ(6, 7) = 0.0000E+00
 AIJ(6, 8) = 0.0000E+00
 AIJ(6, 9) = 0.0000E+00
 AIJ(6,10) = 0.0000E+00
 AIJ(6,11) = 0.0000E+00
 AIJ(6,12) = 0.0000E+00
 AIJ(6,13) = 0.0000E+00
 AIJ(6,14) = 0.0000E+00
 AIJ(6,15) = 0.0000E+00
 AIJ(6,16) = 0.0000E+00
 AIJ(6,17) = 0.0000E+00
 AIJ(6,18) = 0.0000E+00
 AIJ(6,19) = 0.0000E+00
 AIJ(6,20) = 0.0000E+00
 AIJ(6,21) = 0.0000E+00
 AIJ(6,22) = 0.0000E+00
 AIJ(7, 1) = 0.0000E+00
 AIJ(7, 2) = 0.0000E+00
 AIJ(7, 3) = 0.0000E+00
 AIJ(7, 4) = 0.0000E+00
 AIJ(7, 5) = 0.0000E+00
 AIJ(7, 6) = 0.0000E+00
 AIJ(7, 7) = 0.0000E+00
 AIJ(7, 8) = 0.0000E+00
 AIJ(7, 9) = 0.0000E+00
 AIJ(7,10) = 0.0000E+00
 AIJ(7,11) = 0.0000E+00
 AIJ(7,12) = 0.0000E+00
 AIJ(7,13) = 0.0000E+00
 AIJ(7,14) = 0.0000E+00
 AIJ(7,15) = 0.0000E+00
 AIJ(7,16) = 0.0000E+00
 AIJ(7,17) = 0.0000E+00
 AIJ(7,18) = 0.0000E+00
 AIJ(7,19) = 0.0000E+00
 AIJ(7,20) = 0.0000E+00
 AIJ(7,21) = 0.0000E+00
 AIJ(7,22) = 0.0000E+00
 AIJ(8, 1) = 0.1000E+01
 AIJ(8, 2) = 0.1000E+01
 AIJ(8, 3) = 0.1000E+01
 AIJ(8, 4) = 0.1000E+01
 AIJ(8, 5) = 0.1000E+01
 AIJ(8, 6) = 0.1000E+01
 AIJ(8, 7) = 0.0000E+00
 AIJ(8, 8) = 0.0000E+00
 AIJ(8, 9) = 0.0000E+00
 AIJ(8,10) = 0.0000E+00
 AIJ(8,11) = 0.0000E+00
 AIJ(8,12) = 0.0000E+00
 AIJ(8,13) = 0.0000E+00

ORIGINAL PAGE IS
OF POOR QUALITY

AIJ(8,14) = 0.0000E+00
AIJ(8,15) = 0.0000E+00
AIJ(8,16) = 0.0000E+00
AIJ(8,17) = 0.0000E+00
AIJ(8,18) = 0.0000E+00
AIJ(8,19) = 0.0000E+00
AIJ(8,20) = 0.0000E+00
AIJ(8,21) = 0.0000E+00
AIJ(8,22) = 0.0000E+00
AIJ(9, 1) = 0.1000E+01
AIJ(9, 2) = 0.1000E+01
AIJ(9, 3) = 0.1000E+01
AIJ(9, 4) = 0.1000E+01
AIJ(9, 5) = 0.1000E+01
AIJ(9, 6) = 0.1000E+01
AIJ(9, 7) = 0.0000E+00
AIJ(9, 8) = 0.0000E+00
AIJ(9, 9) = 0.0000E+00
AIJ(9,10) = 0.0000E+00
AIJ(9,11) = 0.0000E+00
AIJ(9,12) = 0.0000E+00
AIJ(9,13) = 0.0000E+00
AIJ(9,14) = 0.0000E+00
AIJ(9,15) = 0.0000E+00
AIJ(9,16) = 0.0000E+00
AIJ(9,17) = 0.0000E+00
AIJ(9,18) = 0.0000E+00
AIJ(9,19) = 0.0000E+00
AIJ(9,20) = 0.0000E+00
AIJ(9,21) = 0.0000E+00
AIJ(9,22) = 0.0000E+00
AIJ(10, 1) = 0.1000E+01
AIJ(10, 2) = 0.1000E+01
AIJ(10, 3) = 0.1000E+01
AIJ(10, 4) = 0.1000E+01
AIJ(10, 5) = 0.1000E+01
AIJ(10, 6) = 0.1000E+01
AIJ(10, 7) = 0.0000E+00
AIJ(10, 8) = 0.0000E+00
AIJ(10, 9) = 0.0000E+00
AIJ(10,10) = 0.0000E+00
AIJ(10,11) = 0.0000E+00
AIJ(10,12) = 0.0000E+00
AIJ(10,13) = 0.0000E+00
AIJ(10,14) = 0.0000E+00
AIJ(10,15) = 0.0000E+00
AIJ(10,16) = 0.0000E+00
AIJ(10,17) = 0.0000E+00
AIJ(10,18) = 0.0000E+00
AIJ(10,19) = 0.0000E+00
AIJ(10,20) = 0.0000E+00
AIJ(10,21) = 0.0000E+00
AIJ(10,22) = 0.0000E+00

NUIJ(1, 1) =	0	NUPIJ(1, 1) =	0
NUIJ(1, 2) =	0	NUPIJ(1, 2) =	0
NUIJ(1, 3) =	0	NUPIJ(1, 3) =	0
NUIJ(1, 4) =	0	NUPIJ(1, 4) =	0
NUIJ(1, 5) =	1	NUPIJ(1, 5) =	0
NUIJ(1, 6) =	0	NUPIJ(1, 6) =	1
NUIJ(1, 7) =	0	NUPIJ(1, 7) =	0
NUIJ(1, 8) =	0	NUPIJ(1, 8) =	0
NUIJ(1, 9) =	1	NUPIJ(1, 9) =	0
NUIJ(1,10) =	0	NUPIJ(1,10) =	1
NUIJ(1,11) =	0	NUPIJ(1,11) =	0
NUIJ(1,12) =	0	NUPIJ(1,12) =	0
NUIJ(1,13) =	0	NUPIJ(1,13) =	0
NUIJ(1,14) =	0	NUPIJ(1,14) =	0
NUIJ(1,15) =	0	NUPIJ(1,15) =	0
NUIJ(1,16) =	0	NUPIJ(1,16) =	0
NUIJ(1,17) =	0	NUPIJ(1,17) =	0
NUIJ(1,18) =	0	NUPIJ(1,18) =	0
NUIJ(1,19) =	0	NUPIJ(1,19) =	0
NUIJ(1,20) =	0	NUPIJ(1,20) =	0
NUIJ(1,21) =	0	NUPIJ(1,21) =	0
NUIJ(1,22) =	0	NUPIJ(1,22) =	0
NUIJ(2, 1) =	1	NUPIJ(2, 1) =	0
NUIJ(2, 2) =	0	NUPIJ(2, 2) =	1

NUIJ(2, 3) =	0	NUPIJ(2, 3) =	0
NUIJ(2, 4) =	0	NUPIJ(2, 4) =	0
NUIJ(2, 5) =	0	NUPIJ(2, 5) =	0
NUIJ(2, 6) =	0	NUPIJ(2, 6) =	0
NUIJ(2, 7) =	1	NUPIJ(2, 7) =	0
NUIJ(2, 8) =	0	NUPIJ(2, 8) =	1
NUIJ(2, 9) =	0	NUPIJ(2, 9) =	0
NUIJ(2, 10) =	0	NUPIJ(2, 10) =	0
NUIJ(2, 11) =	0	NUPIJ(2, 11) =	0
NUIJ(2, 12) =	0	NUPIJ(2, 12) =	0
NUIJ(2, 13) =	0	NUPIJ(2, 13) =	0
NUIJ(2, 14) =	0	NUPIJ(2, 14) =	0
NUIJ(2, 15) =	0	NUPIJ(2, 15) =	0
NUIJ(2, 16) =	0	NUPIJ(2, 16) =	0
NUIJ(2, 17) =	0	NUPIJ(2, 17) =	0
NUIJ(2, 18) =	0	NUPIJ(2, 18) =	0
NUIJ(2, 19) =	0	NUPIJ(2, 19) =	0
NUIJ(2, 20) =	0	NUPIJ(2, 20) =	0
NUIJ(2, 21) =	0	NUPIJ(2, 21) =	0
NUIJ(2, 22) =	0	NUPIJ(2, 22) =	0
NUIJ(3, 1) =	0	NUPIJ(3, 1) =	0
NUIJ(3, 2) =	0	NUPIJ(3, 2) =	0
NUIJ(3, 3) =	0	NUPIJ(3, 3) =	1
NUIJ(3, 4) =	1	NUPIJ(3, 4) =	0
NUIJ(3, 5) =	0	NUPIJ(3, 5) =	2
NUIJ(3, 6) =	2	NUPIJ(3, 6) =	0
NUIJ(3, 7) =	1	NUPIJ(3, 7) =	0
NUIJ(3, 8) =	0	NUPIJ(3, 8) =	1
NUIJ(3, 9) =	0	NUPIJ(3, 9) =	1
NUIJ(3, 10) =	1	NUPIJ(3, 10) =	0
NUIJ(3, 11) =	0	NUPIJ(3, 11) =	1
NUIJ(3, 12) =	1	NUPIJ(3, 12) =	0
NUIJ(3, 13) =	2	NUPIJ(3, 13) =	0
NUIJ(3, 14) =	0	NUPIJ(3, 14) =	2
NUIJ(3, 15) =	2	NUPIJ(3, 15) =	1
NUIJ(3, 16) =	1	NUPIJ(3, 16) =	2
NUIJ(3, 17) =	1	NUPIJ(3, 17) =	0
NUIJ(3, 18) =	0	NUPIJ(3, 18) =	1
NUIJ(3, 19) =	1	NUPIJ(3, 19) =	0
NUIJ(3, 20) =	0	NUPIJ(3, 20) =	1
NUIJ(3, 21) =	0	NUPIJ(3, 21) =	0
NUIJ(3, 22) =	0	NUPIJ(3, 22) =	0
NUIJ(4, 1) =	0	NUPIJ(4, 1) =	2
NUIJ(4, 2) =	2	NUPIJ(4, 2) =	0
NUIJ(4, 3) =	0	NUPIJ(4, 3) =	1
NUIJ(4, 4) =	1	NUPIJ(4, 4) =	0
NUIJ(4, 5) =	0	NUPIJ(4, 5) =	0
NUIJ(4, 6) =	0	NUPIJ(4, 6) =	0
NUIJ(4, 7) =	0	NUPIJ(4, 7) =	1
NUIJ(4, 8) =	1	NUPIJ(4, 8) =	0
NUIJ(4, 9) =	1	NUPIJ(4, 9) =	0
NUIJ(4, 10) =	0	NUPIJ(4, 10) =	1
NUIJ(4, 11) =	0	NUPIJ(4, 11) =	1
NUIJ(4, 12) =	1	NUPIJ(4, 12) =	0
NUIJ(4, 13) =	0	NUPIJ(4, 13) =	0
NUIJ(4, 14) =	0	NUPIJ(4, 14) =	0
NUIJ(4, 15) =	0	NUPIJ(4, 15) =	0
NUIJ(4, 16) =	0	NUPIJ(4, 16) =	0
NUIJ(4, 17) =	0	NUPIJ(4, 17) =	0
NUIJ(4, 18) =	0	NUPIJ(4, 18) =	0
NUIJ(4, 19) =	0	NUPIJ(4, 19) =	0
NUIJ(4, 20) =	0	NUPIJ(4, 20) =	0
NUIJ(4, 21) =	1	NUPIJ(4, 21) =	0
NUIJ(4, 22) =	0	NUPIJ(4, 22) =	1
NUIJ(5, 1) =	0	NUPIJ(5, 1) =	0
NUIJ(5, 2) =	0	NUPIJ(5, 2) =	0
NUIJ(5, 3) =	1	NUPIJ(5, 3) =	0
NUIJ(5, 4) =	0	NUPIJ(5, 4) =	1
NUIJ(5, 5) =	0	NUPIJ(5, 5) =	0
NUIJ(5, 6) =	0	NUPIJ(5, 6) =	0
NUIJ(5, 7) =	0	NUPIJ(5, 7) =	1
NUIJ(5, 8) =	1	NUPIJ(5, 8) =	0
NUIJ(5, 9) =	0	NUPIJ(5, 9) =	1
NUIJ(5, 10) =	1	NUPIJ(5, 10) =	0
NUIJ(5, 11) =	0	NUPIJ(5, 11) =	0
NUIJ(5, 12) =	0	NUPIJ(5, 12) =	0
NUIJ(5, 13) =	0	NUPIJ(5, 13) =	0
NUIJ(5, 14) =	0	NUPIJ(5, 14) =	0
NUIJ(5, 15) =	0	NUPIJ(5, 15) =	0
NUIJ(5, 16) =	0	NUPIJ(5, 16) =	0

ORIGINAL PAGE IS
OF POOR QUALITY

NUIJ(5, 17) =	0	NUPIJ(5, 17) =	0
NUIJ(5, 18) =	0	NUPIJ(5, 18) =	0
NUIJ(5, 19) =	0	NUPIJ(5, 19) =	0
NUIJ(5, 20) =	0	NUPIJ(5, 20) =	0
NUIJ(5, 21) =	0	NUPIJ(5, 21) =	0
NUIJ(5, 22) =	0	NUPIJ(5, 22) =	0
NUIJ(6, 1) =	0	NUPIJ(6, 1) =	0
NUIJ(6, 2) =	0	NUPIJ(6, 2) =	0
NUIJ(6, 3) =	0	NUPIJ(6, 3) =	0
NUIJ(6, 4) =	0	NUPIJ(6, 4) =	0
NUIJ(6, 5) =	0	NUPIJ(6, 5) =	0
NUIJ(6, 6) =	0	NUPIJ(6, 6) =	0
NUIJ(6, 7) =	0	NUPIJ(6, 7) =	0
NUIJ(6, 8) =	0	NUPIJ(6, 8) =	0
NUIJ(6, 9) =	0	NUPIJ(6, 9) =	0
NUIJ(6, 10) =	0	NUPIJ(6, 10) =	0
NUIJ(6, 11) =	1	NUPIJ(6, 11) =	0
NUIJ(6, 12) =	0	NUPIJ(6, 12) =	1
NUIJ(6, 13) =	0	NUPIJ(6, 13) =	0
NUIJ(6, 14) =	0	NUPIJ(6, 14) =	0
NUIJ(6, 15) =	0	NUPIJ(6, 15) =	0
NUIJ(6, 16) =	0	NUPIJ(6, 16) =	0
NUIJ(6, 17) =	0	NUPIJ(6, 17) =	0
NUIJ(6, 18) =	0	NUPIJ(6, 18) =	0
NUIJ(6, 19) =	0	NUPIJ(6, 19) =	0
NUIJ(6, 20) =	0	NUPIJ(6, 20) =	0
NUIJ(6, 21) =	0	NUPIJ(6, 21) =	0
NUIJ(6, 22) =	0	NUPIJ(6, 22) =	0
NUIJ(7, 1) =	0	NUPIJ(7, 1) =	0
NUIJ(7, 2) =	0	NUPIJ(7, 2) =	0
NUIJ(7, 3) =	0	NUPIJ(7, 3) =	0
NUIJ(7, 4) =	0	NUPIJ(7, 4) =	0
NUIJ(7, 5) =	0	NUPIJ(7, 5) =	0
NUIJ(7, 6) =	0	NUPIJ(7, 6) =	0
NUIJ(7, 7) =	0	NUPIJ(7, 7) =	0
NUIJ(7, 8) =	0	NUPIJ(7, 8) =	0
NUIJ(7, 9) =	0	NUPIJ(7, 9) =	C
NUIJ(7, 10) =	0	NUPIJ(7, 10) =	0
NUIJ(7, 11) =	1	NUPIJ(7, 11) =	0
NUIJ(7, 12) =	0	NUPIJ(7, 12) =	1
NUIJ(7, 13) =	0	NUPIJ(7, 13) =	1
NUIJ(7, 14) =	1	NUPIJ(7, 14) =	0
NUIJ(7, 15) =	0	NUPIJ(7, 15) =	1
NUIJ(7, 16) =	1	NUPIJ(7, 16) =	0
NUIJ(7, 17) =	0	NUPIJ(7, 17) =	1
NUIJ(7, 18) =	1	NUPIJ(7, 18) =	0
NUIJ(7, 19) =	1	NUPIJ(7, 19) =	2
NUIJ(7, 20) =	2	NUPIJ(7, 20) =	1
NUIJ(7, 21) =	1	NUPIJ(7, 21) =	2
NUIJ(7, 22) =	2	NUPIJ(7, 22) =	1
NUIJ(8, 1) =	0	NUPIJ(8, 1) =	0
NUIJ(8, 2) =	0	NUPIJ(8, 2) =	0
NUIJ(8, 3) =	0	NUPIJ(8, 3) =	0
NUIJ(8, 4) =	0	NUPIJ(8, 4) =	0
NUIJ(8, 5) =	0	NUPIJ(8, 5) =	0
NUIJ(8, 6) =	0	NUPIJ(8, 6) =	0
NUIJ(8, 7) =	0	NUPIJ(8, 7) =	0
NUIJ(8, 8) =	0	NUPIJ(8, 8) =	0
NUIJ(8, 9) =	0	NUPIJ(8, 9) =	0
NUIJ(8, 10) =	0	NUPIJ(8, 10) =	0
NUIJ(8, 11) =	0	NUPIJ(8, 11) =	0
NUIJ(8, 12) =	0	NUPIJ(8, 12) =	0
NUIJ(8, 13) =	0	NUPIJ(8, 13) =	1
NUIJ(8, 14) =	1	NUPIJ(8, 14) =	0
NUIJ(8, 15) =	0	NUPIJ(8, 15) =	0
NUIJ(8, 16) =	0	NUPIJ(8, 16) =	0
NUIJ(8, 17) =	0	NUPIJ(8, 17) =	0
NUIJ(8, 18) =	0	NUPIJ(8, 18) =	0
NUIJ(8, 19) =	0	NUPIJ(8, 19) =	0
NUIJ(8, 20) =	0	NUPIJ(8, 20) =	0
NUIJ(8, 21) =	0	NUPIJ(8, 21) =	0
NUIJ(8, 22) =	0	NUPIJ(8, 22) =	0
NUIJ(9, 1) =	0	NUPIJ(9, 1) =	0
NUIJ(9, 2) =	0	NUPIJ(9, 2) =	0
NUIJ(9, 3) =	0	NUPIJ(9, 3) =	0
NUIJ(9, 4) =	0	NUPIJ(9, 4) =	0
NUIJ(9, 5) =	0	NUPIJ(9, 5) =	0
NUIJ(9, 6) =	0	NUPIJ(9, 6) =	0
NUIJ(9, 7) =	0	NUPIJ(9, 7) =	0
NUIJ(9, 8) =	0	NUPIJ(9, 8) =	0

NUIJ(9, 9) =	0	NUPIJ(9, 9) =	0
NUIJ(9,10) =	0	NUPIJ(9,10) =	0
NUIJ(9,11) =	0	NUPIJ(9,11) =	0
NUIJ(9,12) =	0	NUPIJ(9,12) =	0
NUIJ(9,13) =	0	NUPIJ(9,13) =	0
NUIJ(9,14) =	0	NUPIJ(9,14) =	0
NUIJ(9,15) =	0	NUPIJ(9,15) =	1
NUIJ(9,16) =	1	NUPIJ(9,16) =	0
NUIJ(9,17) =	1	NUPIJ(9,17) =	2
NUIJ(9,18) =	2	NUPIJ(9,18) =	1
NUIJ(9,19) =	0	NUPIJ(9,19) =	1
NUIJ(9,20) =	1	NUPIJ(9,20) =	0
NUIJ(9,21) =	0	NUPIJ(9,21) =	0
NUIJ(9,22) =	0	NUPIJ(9,22) =	0
NUIJ(10, 1) =	0	NUPIJ(10, 1) =	0
NUIJ(10, 2) =	0	NUPIJ(10, 2) =	0
NUIJ(10, 3) =	0	NUPIJ(10, 3) =	0
NUIJ(10, 4) =	0	NUPIJ(10, 4) =	0
NUIJ(10, 5) =	0	NUPIJ(10, 5) =	0
NUIJ(10, 6) =	0	NUPIJ(10, 6) =	0
NUIJ(10, 7) =	0	NUPIJ(10, 7) =	0
NUIJ(10, 8) =	0	NUPIJ(10, 8) =	0
NUIJ(10, 9) =	0	NUPIJ(10, 9) =	0
NUIJ(10,10) =	0	NUPIJ(10,10) =	0
NUIJ(10,11) =	0	NUPIJ(10,11) =	0
NUIJ(10,12) =	0	NUPIJ(10,12) =	0
NUIJ(10,13) =	0	NUPIJ(10,13) =	0
NUIJ(10,14) =	0	NUPIJ(10,14) =	0
NUIJ(10,15) =	0	NUPIJ(10,15) =	0
NUIJ(10,16) =	0	NUPIJ(10,16) =	0
NUIJ(10,17) =	0	NUPIJ(10,17) =	0
NUIJ(10,18) =	0	NUPIJ(10,18) =	0
NUIJ(10,19) =	0	NUPIJ(10,19) =	0
NUIJ(10,20) =	0	NUPIJ(10,20) =	0
NUIJ(10,21) =	0	NUPIJ(10,21) =	1
NUIJ(10,22) =	1	NUPIJ(10,22) =	0
NUIJ(11, 1) =	1	NUPIJ(11, 1) =	0
NUIJ(11, 2) =	1	NUPIJ(11, 2) =	0
NUIJ(11, 3) =	1	NUPIJ(11, 3) =	0
NUIJ(11, 4) =	1	NUPIJ(11, 4) =	0
NUIJ(11, 5) =	1	NUPIJ(11, 5) =	0
NUIJ(11, 6) =	1	NUPIJ(11, 6) =	0
NUIJ(11, 7) =	0	NUPIJ(11, 7) =	0
NUIJ(11, 8) =	0	NUPIJ(11, 8) =	0
NUIJ(11, 9) =	0	NUPIJ(11, 9) =	0
NUIJ(11,10) =	0	NUPIJ(11,10) =	0
NUIJ(11,11) =	0	NUPIJ(11,11) =	0
NUIJ(11,12) =	0	NUPIJ(11,12) =	0
NUIJ(11,13) =	0	NUPIJ(11,13) =	0
NUIJ(11,14) =	0	NUPIJ(11,14) =	0
NUIJ(11,15) =	0	NUPIJ(11,15) =	0
NUIJ(11,16) =	0	NUPIJ(11,16) =	0
NUIJ(11,17) =	0	NUPIJ(11,17) =	0
NUIJ(11,18) =	0	NUPIJ(11,18) =	0
NUIJ(11,19) =	0	NUPIJ(11,19) =	0
NUIJ(11,20) =	0	NUPIJ(11,20) =	0
NUIJ(11,21) =	0	NUPIJ(11,21) =	0
NUIJ(11,22) =	0	NUPIJ(11,22) =	0

SIGIK(1, 1) =	0.1636E+03	ALPIK(1, 1) =	0.5675E-03	BETAIK(1, 1) =	-.9952E-01
SIGIK(1, 2) =	0.1793E+03	ALPIK(1, 2) =	0.1443E-06	BETAIK(1, 2) =	0.3718E+00
SIGIK(1, 3) =	0.0000E+00	ALPIK(1, 3) =	0.0000E+00	BETAIK(1, 3) =	0.0000E+00
SIGIK(1, 4) =	0.0000E+00	ALPIK(1, 4) =	0.0000E+00	BETAIK(1, 4) =	0.0000E+00
SIGIK(1, 5) =	0.1353E+03	ALPIK(1, 5) =	0.1513E-03	BETAIK(1, 5) =	-.2226E+00
SIGIK(1, 6) =	0.1353E+03	ALPIK(1, 6) =	0.1513E-03	BETAIK(1, 6) =	-.2226E+00
SIGIK(1, 7) =	0.0000E+00	ALPIK(1, 7) =	0.0000E+00	BETAIK(1, 7) =	0.0000E+00
SIGIK(1, 8) =	0.1636E+03	ALPIK(1, 8) =	0.5675E-03	BETAIK(1, 8) =	-.9952E-01
SIGIK(1, 9) =	0.0000E+00	ALPIK(1, 9) =	0.0000E+00	BETAIK(1, 9) =	0.0000E+00
SIGIK(1,10) =	0.0000E+00	ALPIK(1,10) =	0.0000E+00	BETAIK(1,10) =	0.0000E+00
SIGIK(2, 1) =	0.1636E+03	ALPIK(2, 1) =	0.5675E-03	BETAIK(2, 1) =	-.9952E-01
SIGIK(2, 2) =	0.1793E+03	ALPIK(2, 2) =	0.1443E-06	BETAIK(2, 2) =	0.3718E+00
SIGIK(2, 3) =	0.0000E+00	ALPIK(2, 3) =	0.0000E+00	BETAIK(2, 3) =	0.0000E+00
SIGIK(2, 4) =	0.0000E+00	ALPIK(2, 4) =	0.0000E+00	BETAIK(2, 4) =	0.0000E+00
SIGIK(2, 5) =	0.1353E+03	ALPIK(2, 5) =	0.1513E-03	BETAIK(2, 5) =	-.2226E+00
SIGIK(2, 6) =	0.1353E+03	ALPIK(2, 6) =	0.1513E-03	BETAIK(2, 6) =	-.2226E+00
SIGIK(2, 7) =	0.0000E+00	ALPIK(2, 7) =	0.0000E+00	BETAIK(2, 7) =	0.0000E+00
SIGIK(2, 8) =	0.1636E+03	ALPIK(2, 8) =	0.5675E-03	BETAIK(2, 8) =	-.9952E-01
SIGIK(2, 9) =	0.0000E+00	ALPIK(2, 9) =	0.0000E+00	BETAIK(2, 9) =	0.0000E+00

670

SIGIK(10,10) = 0.0000E+00 ALPIK(10,10) = 0.0000E+00 BETAIK(10,10) = 0.0000E+00

***** END INPUT *****

+++ SHOCK GEOMETRY +++

X	ZS	RS	RC	CDST	SINT	
0.0000E+00	0.0000E+00	0.0000E+00	0.2300E+03	0.0000E+00	0.1000E+01	1
0.1667E+00	0.1554E-03	0.1667E+00	0.2300E+03	0.7246E-03	0.1000E+01	2
0.3333E+00	0.3108E-03	0.3333E+00	0.2300E+03	0.1449E-02	0.1000E+01	3
0.5000E+00	0.5599E-03	0.5000E+00	0.2300E+03	0.2174E-02	0.1000E+01	4
0.1000E+01	0.2178E-02	0.1000E+01	0.2300E+03	0.4348E-02	0.1000E+01	5
0.1500E+01	0.4892E-02	0.1500E+01	0.2301E+03	0.6521E-02	0.1000E+01	6
0.2000E+01	0.8696E-02	0.2000E+01	0.2301E+03	0.8694E-02	0.1000E+01	7
0.2500E+01	0.1358E-01	0.2500E+01	0.2302E+03	0.1087E-01	0.9999E+00	8
0.3000E+01	0.1956E-01	0.3000E+01	0.2303E+03	0.1304E-01	0.9999E+00	9
0.3500E+01	0.2662E-01	0.3500E+01	0.2304E+03	0.1521E-01	0.9999E+00	10
0.4000E+01	0.3477E-01	0.4000E+01	0.2305E+03	0.1738E-01	0.9998E+00	11
0.4500E+01	0.4400E-01	0.4500E+01	0.2307E+03	0.1955E-01	0.9998E+00	12
0.5000E+01	0.5431E-01	0.5000E+01	0.2308E+03	0.2171E-01	0.9998E+00	13
0.5500E+01	0.6571E-01	0.5499E+01	0.2310E+03	0.2388E-01	0.9997E+00	14
0.6000E+01	0.7819E-01	0.5999E+01	0.2312E+03	0.2604E-01	0.9997E+00	15
0.6500E+01	0.9175E-01	0.6499E+01	0.2314E+03	0.2820E-01	0.9996E+00	16
0.7000E+01	0.1064E+00	0.6999E+01	0.2316E+03	0.3036E-01	0.9995E+00	17
0.7500E+01	0.1221E+00	0.7499E+01	0.2318E+03	0.3252E-01	0.9995E+00	18
0.8000E+01	0.1389E+00	0.7998E+01	0.2321E+03	0.3467E-01	0.9994E+00	19
0.8500E+01	0.1568E+00	0.8498E+01	0.2324E+03	0.3682E-01	0.9993E+00	20
0.9000E+01	0.1757E+00	0.8998E+01	0.2326E+03	0.3897E-01	0.9992E+00	21
0.9500E+01	0.1958E+00	0.9497E+01	0.2329E+03	0.4112E-01	0.9992E+00	22
0.1000E+02	0.2165E+00	0.9997E+01	0.2333E+03	0.4326E-01	0.9991E+00	23

+++ FREESTREAM QUANTITIES +++

MUINF = 0.2885E+02 RHOINF = 0.2766E-07 AINF = 0.2820E+05
TINF = 0.1971E+03 PINF = 0.1572E+02 VINF = 0.8915E+06
MINF = 0.3161E+02 EIINF = 0.9781E+12 EINF = 0.1420E+10 EEIINF = 0.0000E+00

EVIINF(1) = 0.3378E+03
EVIINF(2) = 0.5740E+05
EVIINF(3) = 0.0000E+00
EVIINF(4) = 0.0000E+00
EVIINF(5) = 0.6999E+04
EVIINF(6) = 0.2747E+03
EVIINF(7) = 0.0000E+00
EVIINF(8) = 0.1184E+04
EVIINF(9) = 0.0000E+00
EVIINF(10) = 0.0000E+00

+++ QUANTITIES BEHIND SHOCK +++

TS	ES	PS	RHOS	US	PSIS	HS
.3040E+05	.3054E+12	.1923E+05	.2195E-06	.0000E+00	.0000E+00	.3931E+12
.3040E+05	.3054E+12	.1923E+05	.2195E-06	.6460E+03	.3425E-03	.3931E+12
.3040E+05	.3054E+12	.1923E+05	.2195E-06	.1292E+04	.1370E-02	.3931E+12
.3040E+05	.3054E+12	.1923E+05	.2195E-06	.1938E+04	.3083E-02	.3931E+12
.3040E+05	.3054E+12	.1923E+05	.2195E-06	.3876E+04	.1233E-01	.3931E+12
.3040E+05	.3054E+12	.1923E+05	.2195E-06	.5813E+04	.2774E-01	.3930E+12
.3040E+05	.3054E+12	.1923E+05	.2195E-06	.7751E+04	.4932E-01	.3930E+12
.3040E+05	.3054E+12	.1923E+05	.2195E-06	.9687E+04	.7707E-01	.3930E+12
.3039E+05	.3054E+12	.1923E+05	.2195E-06	.1162E+05	.1110E+00	.3930E+12
.3039E+05	.3054E+12	.1923E+05	.2195E-06	.1356E+05	.1510E+00	.3930E+12

.3039E+05	.3054E+12	.1923E+05	.2195E-06	.1549E+05	.1973E+00	.3929E+12
.3039E+05	.3053E+12	.1922E+05	.2195E-06	.1742E+05	.2497E+00	.3929E+12
.3039E+05	.3053E+12	.1922E+05	.2195E-06	.1936E+05	.3082E+00	.3929E+12
.3038E+05	.3053E+12	.1922E+05	.2195E-06	.2129E+05	.3729E+00	.3928E+12
.3038E+05	.3052E+12	.1922E+05	.2195E-06	.2321E+05	.4438E+00	.3928E+12
.3038E+05	.3052E+12	.1922E+05	.2195E-06	.2514E+05	.5209E+00	.3928E+12
.3037E+05	.3052E+12	.1921E+05	.2195E-06	.2707E+05	.6040E+00	.3927E+12
.3037E+05	.3051E+12	.1921E+05	.2195E-06	.2899E+05	.6934E+00	.3926E+12
.3037E+05	.3051E+12	.1921E+05	.2195E-06	.3091E+05	.7889E+00	.3926E+12
.3036E+05	.3050E+12	.1920E+05	.2195E-06	.3283E+05	.8905E+00	.3925E+12
.3036E+05	.3050E+12	.1920E+05	.2195E-06	.3474E+05	.9983E+00	.3925E+12
.3035E+05	.3049E+12	.1920E+05	.2195E-06	.3666E+05	.1112E+01	.3924E+12
.3035E+05	.3049E+12	.1919E+05	.2195E-06	.3857E+05	.1232E+01	.3923E+12

XXXXX ITK = 18 NXPST = 20 XXXXX

NO. OF VIB. LEVELS TO DISSOCIATION

NI(1) =	34
NI(2) =	27
NI(3) =	0
NI(4) =	0
NI(5) =	29
NI(6) =	38
NI(7) =	0
NI(8) =	33
NI(9) =	0
NI(10) =	0

IZTERM = 23

COUPLED VIBRATION-DISSOCIATION MODELS

TYPE	NO.
VIB. EQUIL.	0
CVD	1
CVDV	2
CVDV-Preferential	3
PARK	4

SHOCK JUMP CONDITION MODELS

TYPE	NO.
CHEMISTRY FROZEN	1
N2 FROZEN, O2 DISS.	2
N2 AND O2 DISS.	3

MILLIKAN AND WHITE DATA FOR N2 AND N2+ WAS SELECTED

XXXXXXXXXXXXXXXXXX

PSI = 2 IX = 1 SHOCK J COND. = 1 CVD MODEL = 4
X = .1667D+00 H = .3931D+12 MU = .2885E+02 P = .1923E+05 RHO = .2195E-06
U = .6460E+03 T = .3040E+05 E = .3054E+12 CIT = .1221E-03 SUMCI = .1000E+01

CI(1) =	0.7680D+00	CM(1) =	0.7908E+00	EVI(1) =	0.3378D+03
CI(2) =	0.2320D+00	CM(2) =	0.2092E+00	EVI(2) =	0.5740D+05
CI(3) =	0.0000D+00	CM(3) =	0.0000E+00	EVI(3) =	0.0000D+00
CI(4) =	0.0000D+00	CM(4) =	0.0000E+00	EVI(4) =	0.0000D+00
CI(5) =	0.0000D+00	CM(5) =	0.0000E+00	EVI(5) =	0.6999D+04
CI(6) =	0.0000D+00	CM(6) =	0.0000E+00	EVI(6) =	0.2747D+03
CI(7) =	0.0000D+00	CM(7) =	0.0000E+00	EVI(7) =	0.0000D+00
CI(8) =	0.0000D+00	CM(8) =	0.0000E+00	EVI(8) =	0.1184D+04

CI(9) = 0.0000D+00 CM(9) = 0.0000E+00 EVI(9) = 0.0000D+00
CI(10) = 0.0000D+00 CM(10) = 0.0000E+00 EVI(10) = 0.0000D+00

TVI(1) = 0.1971E+03
TVI(2) = 0.1971E+03
TVI(3) = 0.0000E+00
TVI(4) = 0.0000E+00
TVI(5) = 0.1971E+03
TVI(6) = 0.1971E+03
TVI(7) = 0.0000E+00
TVI(8) = 0.1971E+03
TVI(9) = 0.0000E+00
TVI(10) = 0.0000E+00 TE1= 0.1971E+03 ITRTE= 0 W1= 0.0000E+00

PSI = 2 IX = 51 SHOCK J COND. = 1 CVD MODEL = 4
X = .2116D+00 H = .3931D+12 MU = .1826E+02 P = .1923E+05 RHO = .2748E-06
U = .1042E+04 T = .1537E+05 E = .3231E+12 CIT = .9766E-03 SUMCI = .1000E+01

CI(1) = 0.4049D+00	CM(1) = 0.2640E+00	EVI(1) = 0.2333D+11
CI(2) = 0.4839D-03	CM(2) = 0.2762E-03	EVI(2) = 0.3649D+11
CI(3) = 0.3542D+00	CM(3) = 0.4617E+00	EVI(3) = 0.0000D+00
CI(4) = 0.2268D+00	CM(4) = 0.2589E+00	EVI(4) = 0.0000D+00
CI(5) = 0.6831D-02	CM(5) = 0.4157E-02	EVI(5) = 0.3920D+11
CI(6) = 0.1520D-02	CM(6) = 0.9251E-03	EVI(6) = 0.3827D+11
CI(7) = 0.1642D-06	CM(7) = 0.5467E-02	EVI(7) = 0.0000D+00
CI(8) = 0.3402D-02	CM(8) = 0.2218E-02	EVI(8) = 0.4235D+11
CI(9) = 0.1588D-02	CM(9) = 0.2070E-02	EVI(9) = 0.0000D+00
CI(10) = 0.2232D-03	CM(10) = 0.2548E-03	EVI(10) = 0.0000D+00

TVI(1) = 0.9457E+04
TVI(2) = 0.1515E+05
TVI(3) = 0.0000E+00
TVI(4) = 0.0000E+00
TVI(5) = 0.1548E+05
TVI(6) = 0.1546E+05
TVI(7) = 0.0000E+00
TVI(8) = 0.1578E+05
TVI(9) = 0.0000E+00
TVI(10) = 0.0000E+00 TE1= 0.1154E+05 ITRTE= 4 W1= 0.198E+28

PSI = 2 IX = 101 SHOCK J COND. = 1 CVD MODEL = 4
X = .3034D+00 H = .3931D+12 MU = .1683E+02 P = .1923E+05 RHO = .3193E-06
U = .1068E+04 T = .1220E+05 E = .3328E+12 CIT = .1953E-02 SUMCI = .1000E+01

CI(1) = 0.2824D+00	CM(1) = 0.1697E+00	EVI(1) = 0.2061D+11
CI(2) = 0.1566D-03	CM(2) = 0.8238E-04	EVI(2) = 0.2872D+11
CI(3) = 0.4789D+00	CM(3) = 0.5755E+00	EVI(3) = 0.0000D+00
CI(4) = 0.2280D+00	CM(4) = 0.2399E+00	EVI(4) = 0.0000D+00
CI(5) = 0.4248D-02	CM(5) = 0.2383E-02	EVI(5) = 0.3023D+11
CI(6) = 0.5445D-03	CM(6) = 0.3055E-03	EVI(6) = 0.2935D+11
CI(7) = 0.2012D-06	CM(7) = 0.6175E-02	EVI(7) = 0.0000D+00
CI(8) = 0.1191D-02	CM(8) = 0.7158E-03	EVI(8) = 0.3230D+11
CI(9) = 0.3220D-02	CM(9) = 0.3870E-02	EVI(9) = 0.0000D+00
CI(10) = 0.1220D-02	CM(10) = 0.1284E-02	EVI(10) = 0.0000D+00

TVI(1) = 0.8528E+04
TVI(2) = 0.1215E+05
TVI(3) = 0.0000E+00
TVI(4) = 0.0000E+00
TVI(5) = 0.1223E+05
TVI(6) = 0.1222E+05
TVI(7) = 0.0000E+00
TVI(8) = 0.1238E+05
TVI(9) = 0.0000E+00
TVI(10) = 0.0000E+00 TE1= 0.1082E+05 ITRTE= 4 W1= 0.134E+28

PSI = 2 IX = 151 SHOCK J COND. = 1 CVD MODEL = 4
X = .4010D+00 H = .3931D+12 MU = .1632E+02 P = .1923E+05 RHO = .3440E-06

U = .1113E+04 T = .1097E+05 E = .3372E+12 CIT = .1953E-02 SUMCI = .1000E+01

CI(1) = 0.2333D+00	CM(1) = 0.1359E+00	EVI(1) = 0.2090D+11
CI(2) = 0.1017D-03	CM(2) = 0.5189E-04	EVI(2) = 0.2561D+11
CI(3) = 0.5284D+00	CM(3) = 0.6155E+00	EVI(3) = 0.0000D+00
CI(4) = 0.2278D+00	CM(4) = 0.2323E+00	EVI(4) = 0.0000D+00
CI(5) = 0.3675D-02	CM(5) = 0.1999E-02	EVI(5) = 0.2682D+11
CI(6) = 0.2822D-03	CM(6) = 0.1535E-03	EVI(6) = 0.2595D+11
CI(7) = 0.2388D-06	CM(7) = 0.7103E-02	EVI(7) = 0.0000D+00
CI(8) = 0.5677D-03	CM(8) = 0.3307E-03	EVI(8) = 0.2853D+11
CI(9) = 0.3940D-02	CM(9) = 0.4590E-02	EVI(9) = 0.0000D+00
CI(10) = 0.1989D-02	CM(10) = 0.2029E-02	EVI(10) = 0.0000D+00

TVI(1) = 0.8629E+04
TVI(2) = 0.1095E+05
TVI(3) = 0.0000E+00
TVI(4) = 0.0000E+00
TVI(5) = 0.1099E+05
TVI(6) = 0.1099E+05
TVI(7) = 0.0000E+00
TVI(8) = 0.1110E+05
TVI(9) = 0.0000E+00
TVI(10) = 0.0000E+00 TE1= 0.1029E+05 ITRTE= 4 W1= 0.958E+27

PSI = 2 IX = 201 SHOCK J COND. = 1 CVD MODEL = 4

X = .4987D+00 H = .3931D+12 MU = .1604E+02 P = .1923E+05 RHO = .3601E-06

U = .1205E+04 T = .1030E+05 E = .3397E+12 CIT = .1953E-02 SUMCI = .1000E+01

CI(1) = 0.2054D+00	CM(1) = 0.1176E+00	EVI(1) = 0.2102D+11
CI(2) = 0.7790D-04	CM(2) = 0.3904E-04	EVI(2) = 0.2389D+11
CI(3) = 0.5562D+00	CM(3) = 0.6369E+00	EVI(3) = 0.0000D+00
CI(4) = 0.2275D+00	CM(4) = 0.2280E+00	EVI(4) = 0.0000D+00
CI(5) = 0.3341D-02	CM(5) = 0.1786E-02	EVI(5) = 0.2496D+11
CI(6) = 0.1848D-03	CM(6) = 0.9880E-04	EVI(6) = 0.2410D+11
CI(7) = 0.2674D-06	CM(7) = 0.7816E-02	EVI(7) = 0.0000D+00
CI(8) = 0.3429D-03	CM(8) = 0.1963E-03	EVI(8) = 0.2647D+11
CI(9) = 0.4344D-02	CM(9) = 0.4974E-02	EVI(9) = 0.0000D+00
CI(10) = 0.2541D-02	CM(10) = 0.2547E-02	EVI(10) = 0.0000D+00

TVI(1) = 0.8670E+04
TVI(2) = 0.1029E+05
TVI(3) = 0.0000E+00
TVI(4) = 0.0000E+00
TVI(5) = 0.1032E+05
TVI(6) = 0.1031E+05
TVI(7) = 0.0000E+00
TVI(8) = 0.1040E+05
TVI(9) = 0.0000E+00
TVI(10) = 0.0000E+00 TE1= 0.9903E+04 ITRTE= 3 W1= 0.674E+27

PSI = 2 IX = 202 SHOCK J COND. = 1 CVD MODEL = 4

X = .5007D+00 H = .3931D+12 MU = .1604E+02 P = .1923E+05 RHO = .3604E-06

U = .1207E+04 T = .1029E+05 E = .3397E+12 CIT = .1953E-02 SUMCI = .1000E+01

CI(1) = 0.2050D+00	CM(1) = 0.1173E+00	EVI(1) = 0.2102D+11
CI(2) = 0.7754D-04	CM(2) = 0.3886E-04	EVI(2) = 0.2386D+11
CI(3) = 0.5567D+00	CM(3) = 0.6372E+00	EVI(3) = 0.0000D+00
CI(4) = 0.2275D+00	CM(4) = 0.2280E+00	EVI(4) = 0.0000D+00
CI(5) = 0.3335D-02	CM(5) = 0.1782E-02	EVI(5) = 0.2493D+11
CI(6) = 0.1836D-03	CM(6) = 0.9810E-04	EVI(6) = 0.2407D+11
CI(7) = 0.2678D-06	CM(7) = 0.7828E-02	EVI(7) = 0.0000D+00
CI(8) = 0.3401D-03	CM(8) = 0.1947E-03	EVI(8) = 0.2643D+11
CI(9) = 0.4350D-02	CM(9) = 0.4980E-02	EVI(9) = 0.0000D+00
CI(10) = 0.2550D-02	CM(10) = 0.2556E-02	EVI(10) = 0.0000D+00

TVI(1) = 0.8670E+04
TVI(2) = 0.1028E+05
TVI(3) = 0.0000E+00
TVI(4) = 0.0000E+00
TVI(5) = 0.1031E+05
TVI(6) = 0.1030E+05

TVI(7) = 0.0000E+00
 TVI(8) = 0.1039E+05
 TVI(9) = 0.0000E+00
 TVI(10) = 0.0000E+00 TE1= 0.9897E+04 ITRTE= 3 W1= 0.670E+27

XXXXXXXXXXXXXXXXXX

PSI = 3 IX = 1 SHOCK J COND. = 1 CVD MODEL = 4
 X = .3333D+00 H = .3931D+12 MU = .2885E+02 P = .1923E+05 RHO = .2195E-06
 U = .1292E+04 T = .3040E+05 E = .3054E+12 CIT = .1221E-03 SUMCI = .1000E+01

CI(1) = 0.7680D+00	CM(1) = 0.7908E+00	EVI(1) = 0.3378D+03
CI(2) = 0.2320D+00	CM(2) = 0.2092E+00	EVI(2) = 0.5740D+05
CI(3) = 0.0000D+00	CM(3) = 0.0000E+00	EVI(3) = 0.0000D+00
CI(4) = 0.0000D+00	CM(4) = 0.0000E+00	EVI(4) = 0.0000D+00
CI(5) = 0.0000D+00	CM(5) = 0.0000E+00	EVI(5) = 0.6999D+04
CI(6) = 0.0000D+00	CM(6) = 0.0000E+00	EVI(6) = 0.2747D+03
CI(7) = 0.0000D+00	CM(7) = 0.0000E+00	EVI(7) = 0.0000D+00
CI(8) = 0.0000D+00	CM(8) = 0.0000E+00	EVI(8) = 0.1184D+04
CI(9) = 0.0000D+00	CM(9) = 0.0000E+00	EVI(9) = 0.0000D+00
CI(10) = 0.0000D+00	CM(10) = 0.0000E+00	EVI(10) = 0.0000D+00

TVI(1) = 0.1971E+03
 TVI(2) = 0.1971E+03
 TVI(3) = 0.0000E+00
 TVI(4) = 0.0000E+00
 TVI(5) = 0.1971E+03
 TVI(6) = 0.1971E+03
 TVI(7) = 0.0000E+00
 TVI(8) = 0.1971E+03
 TVI(9) = 0.0000E+00
 TVI(10) = 0.0000E+00 TE1= 0.1971E+03 ITRTE= 3 W1= 0.670E+27

PSI = 3 IX = 51 SHOCK J COND. = 1 CVD MODEL = 4
 X = .4008D+00 H = .3931D+12 MU = .1815E+02 P = .1923E+05 RHO = .2774E-06
 U = .1443E+04 T = .1513E+05 E = .3237E+12 CIT = .1953E-02 SUMCI = .1000E+01

CI(1) = 0.3957D+00	CM(1) = 0.2563E+00	EVI(1) = 0.2299D+11
CI(2) = 0.4442D-03	CM(2) = 0.2519E-03	EVI(2) = 0.3591D+11
CI(3) = 0.3636D+00	CM(3) = 0.4711E+00	EVI(3) = 0.0000D+00
CI(4) = 0.2270D+00	CM(4) = 0.2574E+00	EVI(4) = 0.0000D+00
CI(5) = 0.6581D-02	CM(5) = 0.3979E-02	EVI(5) = 0.3850D+11
CI(6) = 0.1474D-02	CM(6) = 0.8915E-03	EVI(6) = 0.3757D+11
CI(7) = 0.1660D-06	CM(7) = 0.5491E-02	EVI(7) = 0.0000D+00
CI(8) = 0.3235D-02	CM(8) = 0.2096E-02	EVI(8) = 0.4156D+11
CI(9) = 0.1702D-02	CM(9) = 0.2204E-02	EVI(9) = 0.0000D+00
CI(10) = 0.2644D-03	CM(10) = 0.2998E-03	EVI(10) = 0.0000D+00

TVI(1) = 0.9340E+04
 TVI(2) = 0.1493E+05
 TVI(3) = 0.0000E+00
 TVI(4) = 0.0000E+00
 TVI(5) = 0.1522E+05
 TVI(6) = 0.1521E+05
 TVI(7) = 0.0000E+00
 TVI(8) = 0.1552E+05
 TVI(9) = 0.0000E+00
 TVI(10) = 0.0000E+00 TE1= 0.1152E+05 ITRTE= 4 W1= 0.174E+28

PSI = 3 IX = 101 SHOCK J COND. = 1 CVD MODEL = 4
 X = .4985D+00 H = .3931D+12 MU = .1699E+02 P = .1923E+05 RHO = .3130E-06
 U = .1505E+04 T = .1255E+05 E = .3316E+12 CIT = .1953E-02 SUMCI = .1000E+01

CI(1) = 0.2965D+00	CM(1) = 0.1798E+00	EVI(1) = 0.2065D+11
CI(2) = 0.1778D-03	CM(2) = 0.9436E-04	EVI(2) = 0.2961D+11
CI(3) = 0.4647D+00	CM(3) = 0.5635E+00	EVI(3) = 0.0000D+00
CI(4) = 0.2281D+00	CM(4) = 0.2421E+00	EVI(4) = 0.0000D+00
CI(5) = 0.4458D-02	CM(5) = 0.2524E-02	EVI(5) = 0.3123D+11
CI(6) = 0.6427D-03	CM(6) = 0.3638E-03	EVI(6) = 0.3034D+11

CI(7) = 0.1937D-06	CM(7) = 0.5997E-02	EVI(7) = 0.0000D+00
CI(8) = 0.1418D-02	CM(8) = 0.8600E-03	EVI(8) = 0.3341D+11
CI(9) = 0.3022D-02	CM(9) = 0.3665E-02	EVI(9) = 0.0000D+00
CI(10) = 0.1044D-02	CM(10) = 0.1108E-02	EVI(10) = 0.0000D+00

TVI(1) = 0.8543E+04	
TVI(2) = 0.1250E+05	
TVI(3) = 0.0000E+00	
TVI(4) = 0.0000E+00	
TVI(5) = 0.1259E+05	
TVI(6) = 0.1258E+05	
TVI(7) = 0.0000E+00	
TVI(8) = 0.1276E+05	
TVI(9) = 0.0000E+00	
TVI(10) = 0.0000E+00	TE1= 0.1094E+05 ITRTE= 3 W1= 0.138E+28

PSI = 3 IX = 102 SHOCK J COND. = 1 CVD MODEL = 4
X = .5004D+00 H = .3931D+12 MU = .1697E+02 P = .1923E+05 RHO = .3135E-06
U = .1507E+04 T = .1252E+05 E = .3317E+12 CIT = .1953E-02 SUMCI = .1000E+01

CI(1) = 0.2954D+00	CM(1) = 0.1790E+00	EVI(1) = 0.2065D+11
CI(2) = 0.1759D-03	CM(2) = 0.9332E-04	EVI(2) = 0.2954D+11
CI(3) = 0.4658D+00	CM(3) = 0.5645E+00	EVI(3) = 0.0000D+00
CI(4) = 0.2281D+00	CM(4) = 0.2419E+00	EVI(4) = 0.0000D+00
CI(5) = 0.4440D-02	CM(5) = 0.2512E-02	EVI(5) = 0.3115D+11
CI(6) = 0.6342D-03	CM(6) = 0.3587E-03	EVI(6) = 0.3026D+11
CI(7) = 0.1943D-06	CM(7) = 0.6010E-02	EVI(7) = 0.0000D+00
CI(8) = 0.1399D-02	CM(8) = 0.8476E-03	EVI(8) = 0.3331D+11
CI(9) = 0.3038D-02	CM(9) = 0.3682E-02	EVI(9) = 0.0000D+00
CI(10) = 0.1058D-02	CM(10) = 0.1122E-02	EVI(10) = 0.0000D+00

TVI(1) = 0.8541E+04	
TVI(2) = 0.1247E+05	
TVI(3) = 0.0000E+00	
TVI(4) = 0.0000E+00	
TVI(5) = 0.1256E+05	
TVI(6) = 0.1255E+05	
TVI(7) = 0.0000E+00	
TVI(8) = 0.1273E+05	
TVI(9) = 0.0000E+00	
TVI(10) = 0.0000E+00	TE1= 0.1093E+05 ITRTE= 3 W1= 0.138E+28

XXXXXXXXXXXXXXXXXX

-- PSI = 0.000 STREAMLINE --

PSI = 4 IX = 4 SHOCK J COND. = 1 CVD MODEL = 4
X = .5000D+00 H = .3931D+12 MU = .1574E+02 P = .1923E+05 RHO = .3793E-06
U = .1089E+04 T = .9548E+04 E = .3424E+12 CIT = .1221E-03 SUMCI = .1000E+01

CI(1) = 0.1748D+00	CM(1) = 0.9826E-01	EVI(1) = 0.2115D+11
CI(2) = 0.4475D-04	CM(2) = 0.2202E-04	EVI(2) = 0.2197D+11
CI(3) = 0.5869D+00	CM(3) = 0.6597E+00	EVI(3) = 0.0000D+00
CI(4) = 0.2273D+00	CM(4) = 0.2237E+00	EVI(4) = 0.0000D+00
CI(5) = 0.2967D-02	CM(5) = 0.1557E-02	EVI(5) = 0.2286D+11
CI(6) = 0.3337D-04	CM(6) = 0.1751E-04	EVI(6) = 0.2201D+11
CI(7) = 0.2924D-06	CM(7) = 0.8390E-02	EVI(7) = 0.0000D+00
CI(8) = 0.1000D-07	CM(8) = 0.5620E-08	EVI(8) = 0.2414D+11
CI(9) = 0.4788D-02	CM(9) = 0.5381E-02	EVI(9) = 0.0000D+00
CI(10) = 0.3048D-02	CM(10) = 0.2999E-02	EVI(10) = 0.0000D+00

TVI(1) = 0.8713E+04	
TVI(2) = 0.9546E+04	
TVI(3) = 0.0000E+00	
TVI(4) = 0.0000E+00	
TVI(5) = 0.9553E+04	
TVI(6) = 0.9551E+04	
TVI(7) = 0.0000E+00	
TVI(8) = 0.9614E+04	
TVI(9) = 0.0000E+00	
TVI(10) = 0.0000E+00	TE1= 0.8876E+04 ITRTE= 3 W1= 0.991E+28

TVI(7) = 0.0000E+00
 TVI(8) = 0.8540E+04
 TVI(9) = 0.0000E+00
 TVI(10) = 0.0000E+00 TE1= 0.8478E+04 ITRTE= 3 W1= 0.974E+26

XXXXXXXXXXXXXXXXXX

PSI = 6 IX = 1 SHOCK J COND. = 1 CVD MODEL = 4
 X = .1500D+01 H = .3930D+12 MU = .2885E+02 P = .1923E+05 RHO = .2195E-06
 U = .5813E+04 T = .3040E+05 E = .3054E+12 CIT = .1221E-03 SUMCI = .1000E+01

CI(1) = 0.7680D+00	CM(1) = 0.7908E+00	EVI(1) = 0.3378D+03
CI(2) = 0.2320D+00	CM(2) = 0.2092E+00	EVI(2) = 0.5740D+05
CI(3) = 0.0000D+00	CM(3) = 0.0000E+00	EVI(3) = 0.0000D+00
CI(4) = 0.0000D+00	CM(4) = 0.0000E+00	EVI(4) = 0.0000D+00
CI(5) = 0.0000D+00	CM(5) = 0.0000E+00	EVI(5) = 0.6999D+04
CI(6) = 0.0000D+00	CM(6) = 0.0000E+00	EVI(6) = 0.2747D+03
CI(7) = 0.0000D+00	CM(7) = 0.0000E+00	EVI(7) = 0.0000D+00
CI(8) = 0.0000D+00	CM(8) = 0.0000E+00	EVI(8) = 0.1184D+04
CI(9) = 0.0000D+00	CM(9) = 0.0000E+00	EVI(9) = 0.0000D+00
CI(10) = 0.0000D+00	CM(10) = 0.0000E+00	EVI(10) = 0.0000D+00

TVI(1) = 0.1971E+03
 TVI(2) = 0.1971E+03
 TVI(3) = 0.0000E+00
 TVI(4) = 0.0000E+00
 TVI(5) = 0.1971E+03
 TVI(6) = 0.1971E+03
 TVI(7) = 0.0000E+00
 TVI(8) = 0.1971E+03
 TVI(9) = 0.0000E+00
 TVI(10) = 0.0000E+00 TE1= 0.1971E+03 ITRTE= 3 W1= 0.974E+26

PSI = 6 IX = 51 SHOCK J COND. = 1 CVD MODEL = 4
 X = .1754D+01 H = .3930D+12 MU = .1828E+02 P = .1923E+05 RHO = .2743E-06
 U = .6100E+04 T = .1542E+05 E = .3229E+12 CIT = .7813E-02 SUMCI = .1000E+01

CI(1) = 0.4067D+00	CM(1) = 0.2654E+00	EVI(1) = 0.2340D+11
CI(2) = 0.4916D-03	CM(2) = 0.2809E-03	EVI(2) = 0.3659D+11
CI(3) = 0.3524D+00	CM(3) = 0.4600E+00	EVI(3) = 0.0000D+00
CI(4) = 0.2268D+00	CM(4) = 0.2592E+00	EVI(4) = 0.0000D+00
CI(5) = 0.6879D-02	CM(5) = 0.4192E-02	EVI(5) = 0.3933D+11
CI(6) = 0.1526D-02	CM(6) = 0.9296E-03	EVI(6) = 0.3839D+11
CI(7) = 0.1640D-06	CM(7) = 0.5466E-02	EVI(7) = 0.0000D+00
CI(8) = 0.3427D-02	CM(8) = 0.2237E-02	EVI(8) = 0.4249D+11
CI(9) = 0.1574D-02	CM(9) = 0.2055E-02	EVI(9) = 0.0000D+00
CI(10) = 0.2139D-03	CM(10) = 0.2445E-03	EVI(10) = 0.0000D+00

TVI(1) = 0.9479E+04
 TVI(2) = 0.1519E+05
 TVI(3) = 0.0000E+00
 TVI(4) = 0.0000E+00
 TVI(5) = 0.1552E+05
 TVI(6) = 0.1550E+05
 TVI(7) = 0.0000E+00
 TVI(8) = 0.1583E+05
 TVI(9) = 0.0000E+00
 TVI(10) = 0.0000E+00 TE1= 0.1154E+05 ITRTE= 4 W1= 0.203E+28

PSI = 6 IX = 101 SHOCK J COND. = 1 CVD MODEL = 4
 X = .2254D+01 H = .3930D+12 MU = .1692E+02 P = .1923E+05 RHO = .3159E-06
 U = .6705E+04 T = .1238E+05 E = .3322E+12 CIT = .1563E-01 SUMCI = .1000E+01

CI(1) = 0.2900D+00	CM(1) = 0.1751E+00	EVI(1) = 0.2062D+11
CI(2) = 0.1675D-03	CM(2) = 0.8856E-04	EVI(2) = 0.2919D+11
CI(3) = 0.4713D+00	CM(3) = 0.5691E+00	EVI(3) = 0.0000D+00
CI(4) = 0.2281D+00	CM(4) = 0.2411E+00	EVI(4) = 0.0000D+00
CI(5) = 0.4359D-02	CM(5) = 0.2457E-02	EVI(5) = 0.3076D+11
CI(6) = 0.5951D-03	CM(6) = 0.3355E-03	EVI(6) = 0.2987D+11

CI(7) = 0.1971D-06	CM(7) = 0.6077E-02	EVI(7) = 0.0000D+00
CI(8) = 0.1309D-02	CM(8) = 0.7902E-03	EVI(8) = 0.3288D+11
CI(9) = 0.3116D-02	CM(9) = 0.3763E-02	EVI(9) = 0.0000D+00
CI(10) = 0.1124D-02	CM(10) = 0.1188E-02	EVI(10) = 0.0000D+00

TVI(1) = 0.8533E+04	
TVI(2) = 0.1234E+05	
TVI(3) = 0.0000E+00	
TVI(4) = 0.0000E+00	
TVI(5) = 0.1242E+05	
TVI(6) = 0.1241E+05	
TVI(7) = 0.0000E+00	
TVI(8) = 0.1258E+05	
TVI(9) = 0.0000E+00	
TVI(10) = 0.0000E+00	TE1= 0.1089E+05 ITRTE= 3 W1= 0.137E+28

PSI = 6 IX = 151 SHOCK J COND. = 1 CVD MODEL = 4
X = .3036D+01 H = .3930D+12 MU = .1628E+02 P = .1923E+05 RHO = .3460E-06
U = .7698E+04 T = .1088E+05 E = .3375E+12 CIT = .1563E-01 SUMCI = .1000E+01

CI(1) = 0.2297D+00	CM(1) = 0.1335E+00	EVI(1) = 0.2093D+11
CI(2) = 0.9852D-04	CM(2) = 0.5012E-04	EVI(2) = 0.2537D+11
CI(3) = 0.5320D+00	CM(3) = 0.6183E+00	EVI(3) = 0.0000D+00
CI(4) = 0.2278D+00	CM(4) = 0.2318E+00	EVI(4) = 0.0000D+00
CI(5) = 0.3639D-02	CM(5) = 0.1974E-02	EVI(5) = 0.2657D+11
CI(6) = 0.2670D-03	CM(6) = 0.1449E-03	EVI(6) = 0.2570D+11
CI(7) = 0.2424D-06	CM(7) = 0.7194E-02	EVI(7) = 0.0000D+00
CI(8) = 0.5317D-03	CM(8) = 0.3090E-03	EVI(8) = 0.2824D+11
CI(9) = 0.3998D-02	CM(9) = 0.4647E-02	EVI(9) = 0.0000D+00
CI(10) = 0.2057D-02	CM(10) = 0.2094E-02	EVI(10) = 0.0000D+00

TVI(1) = 0.8639E+04	
TVI(2) = 0.1086E+05	
TVI(3) = 0.0000E+00	
TVI(4) = 0.0000E+00	
TVI(5) = 0.1090E+05	
TVI(6) = 0.1090E+05	
TVI(7) = 0.0000E+00	
TVI(8) = 0.1101E+05	
TVI(9) = 0.0000E+00	
TVI(10) = 0.0000E+00	TE1= 0.1024E+05 ITRTE= 4 W1= 0.921E+27

PSI = 6 IX = 201 SHOCK J COND. = 1 CVD MODEL = 4
X = .3817D+01 H = .3930D+12 MU = .1599E+02 P = .1922E+05 RHO = .3632E-06
U = .8752E+04 T = .1018E+05 E = .3401E+12 CIT = .1563E-01 SUMCI = .1000E+01

CI(1) = 0.2004D+00	CM(1) = 0.1144E+00	EVI(1) = 0.2101D+11
CI(2) = 0.7376D-04	CM(2) = 0.3685E-04	EVI(2) = 0.2357D+11
CI(3) = 0.5613D+00	CM(3) = 0.6406E+00	EVI(3) = 0.0000D+00
CI(4) = 0.2275D+00	CM(4) = 0.2273E+00	EVI(4) = 0.0000D+00
CI(5) = 0.3268D-02	CM(5) = 0.1741E-02	EVI(5) = 0.2461D+11
CI(6) = 0.1701D-03	CM(6) = 0.9061E-04	EVI(6) = 0.2376D+11
CI(7) = 0.2732D-06	CM(7) = 0.7963E-02	EVI(7) = 0.0000D+00
CI(8) = 0.3098D-03	CM(8) = 0.1768E-03	EVI(8) = 0.2609D+11
CI(9) = 0.4422D-02	CM(9) = 0.5047E-02	EVI(9) = 0.0000D+00
CI(10) = 0.2651D-02	CM(10) = 0.2649E-02	EVI(10) = 0.0000D+00

TVI(1) = 0.8666E+04	
TVI(2) = 0.1016E+05	
TVI(3) = 0.0000E+00	
TVI(4) = 0.0000E+00	
TVI(5) = 0.1019E+05	
TVI(6) = 0.1019E+05	
TVI(7) = 0.0000E+00	
TVI(8) = 0.1027E+05	
TVI(9) = 0.0000E+00	
TVI(10) = 0.0000E+00	TE1= 0.9823E+04 ITRTE= 3 W1= 0.622E+27

PSI = 6 IX = 251 SHOCK J COND. = 1 CVD MODEL = 4

X = .4598D+01 H = .3930D+12 MU = .1582E+02 P = .1922E+05 RHO = .3744E-06
 U = .9851E+04 T = .9764E+04 E = .3417E+12 CIT = .1563E-01 SUMCI = .1000E+01

CI(1) = 0.1828D+00	CM(1) = 0.1032E+00	EVI(1) = 0.2087D+11
CI(2) = 0.5970D-04	CM(2) = 0.2950E-04	EVI(2) = 0.2250D+11
CI(3) = 0.5789D+00	CM(3) = 0.6536E+00	EVI(3) = 0.0000D+00
CI(4) = 0.2273D+00	CM(4) = 0.2247E+00	EVI(4) = 0.0000D+00
CI(5) = 0.2967D-02	CM(5) = 0.1564E-02	EVI(5) = 0.2346D+11
CI(6) = 0.1270D-03	CM(6) = 0.6696E-04	EVI(6) = 0.2261D+11
CI(7) = 0.2943D-06	CM(7) = 0.8484E-02	EVI(7) = 0.0000D+00
CI(8) = 0.2166D-03	CM(8) = 0.1223E-03	EVI(8) = 0.2481D+11
CI(9) = 0.4676D-02	CM(9) = 0.5279E-02	EVI(9) = 0.0000D+00
CI(10) = 0.3051D-02	CM(10) = 0.3016E-02	EVI(10) = 0.0000D+00

TVI(1) = 0.8618E+04
 TVI(2) = 0.9753E+04
 TVI(3) = 0.0000E+00
 TVI(4) = 0.0000E+00
 TVI(5) = 0.9773E+04
 TVI(6) = 0.9772E+04
 TVI(7) = 0.0000E+00
 TVI(8) = 0.9842E+04
 TVI(9) = 0.0000E+00
 TVI(10) = 0.0000E+00 TE1= 0.9533E+04 ITRTE= 4 W1= 0.455E+29

PSI = 6 IX = 301 SHOCK J COND. = 1 CVD MODEL = 4

X = .5379D+01 H = .3930D+12 MU = .1570E+02 P = .1922E+05 RHO = .3824E-06
 U = .1098E+05 T = .9488E+04 E = .3428E+12 CIT = .1563E-01 SUMCI = .1000E+01

CI(1) = 0.1710D+00	CM(1) = 0.9581E-01	EVI(1) = 0.2069D+11
CI(2) = 0.5064D-04	CM(2) = 0.2485E-04	EVI(2) = 0.2179D+11
CI(3) = 0.5907D+00	CM(3) = 0.6621E+00	EVI(3) = 0.0000D+00
CI(4) = 0.2271D+00	CM(4) = 0.2229E+00	EVI(4) = 0.0000D+00
CI(5) = 0.2725D-02	CM(5) = 0.1426E-02	EVI(5) = 0.2270D+11
CI(6) = 0.1035D-03	CM(6) = 0.5415E-04	EVI(6) = 0.2185D+11
CI(7) = 0.3094D-06	CM(7) = 0.8855E-02	EVI(7) = 0.0000D+00
CI(8) = 0.1678D-03	CM(8) = 0.9405E-04	EVI(8) = 0.2397D+11
CI(9) = 0.4846D-02	CM(9) = 0.5432E-02	EVI(9) = 0.0000D+00
CI(10) = 0.3337D-02	CM(10) = 0.3275E-02	EVI(10) = 0.0000D+00

TVI(1) = 0.8554E+04
 TVI(2) = 0.9479E+04
 TVI(3) = 0.0000E+00
 TVI(4) = 0.0000E+00
 TVI(5) = 0.9496E+04
 TVI(6) = 0.9495E+04
 TVI(7) = 0.0000E+00
 TVI(8) = 0.9555E+04
 TVI(9) = 0.0000E+00
 TVI(10) = 0.0000E+00 TE1= 0.9322E+04 ITRTE= 4 W1= 0.355E+27

PSI = 6 IX = 351 SHOCK J COND. = 1 CVD MODEL = 4

X = .6161D+01 H = .3930D+12 MU = .1562E+02 P = .1921E+05 RHO = .3884E-06
 U = .1214E+05 T = .9290E+04 E = .3435E+12 CIT = .1563E-01 SUMCI = .1000E+01

CI(1) = 0.1624D+00	CM(1) = 0.9055E-01	EVI(1) = 0.2050D+11
CI(2) = 0.4437D-04	CM(2) = 0.2166E-04	EVI(2) = 0.2128D+11
CI(3) = 0.5993D+00	CM(3) = 0.6682E+00	EVI(3) = 0.0000D+00
CI(4) = 0.2271D+00	CM(4) = 0.2216E+00	EVI(4) = 0.0000D+00
CI(5) = 0.2533D-02	CM(5) = 0.1318E-02	EVI(5) = 0.2215D+11
CI(6) = 0.8881D-04	CM(6) = 0.4623E-04	EVI(6) = 0.2131D+11
CI(7) = 0.3207D-06	CM(7) = 0.9132E-02	EVI(7) = 0.0000D+00
CI(8) = 0.1385D-03	CM(8) = 0.7720E-04	EVI(8) = 0.2336D+11
CI(9) = 0.4969D-02	CM(9) = 0.5541E-02	EVI(9) = 0.0000D+00
CI(10) = 0.3552D-02	CM(10) = 0.3468E-02	EVI(10) = 0.0000D+00

TVI(1) = 0.8492E+04
 TVI(2) = 0.9282E+04
 TVI(3) = 0.0000E+00
 TVI(4) = 0.0000E+00

TVI(5) = 0.9297E+04
TVI(6) = 0.9296E+04
TVI(7) = 0.0000E+00
TVI(8) = 0.9349E+04
TVI(9) = 0.0000E+00
TVI(10) = 0.0000E+00 TE1= 0.9162E+04 ITRTE= 4 W1= 0.290E+27

PSI = 6 IX = 401 SHOCK J COND. = 1 CVD MODEL = 4
X = .6942D+01 H = .3930D+12 MU = .1556E+02 P = .1920E+05 RHO = .3931E-06
U = .1332E+05 T = .9141E+04 E = .3441E+12 CIT = .1563E-01 SUMCI = .1000E+01

CI(1) = 0.1559D+00	CM(1) = 0.8659E-01	EVI(1) = 0.2034D+11
CI(2) = 0.3981D-04	CM(2) = 0.1936E-04	EVI(2) = 0.2090D+11
CI(3) = 0.6058D+00	CM(3) = 0.6728E+00	EVI(3) = 0.0000D+00
CI(4) = 0.2270D+00	CM(4) = 0.2207E+00	EVI(4) = 0.0000D+00
CI(5) = 0.2379D-02	CM(5) = 0.1233E-02	EVI(5) = 0.2174D+11
CI(6) = 0.7883D-04	CM(6) = 0.4087E-04	EVI(6) = 0.2090D+11
CI(7) = 0.3296D-06	CM(7) = 0.9346E-02	EVI(7) = 0.0000D+00
CI(8) = 0.1191D-03	CM(8) = 0.6611E-04	EVI(8) = 0.2290D+11
CI(9) = 0.5063D-02	CM(9) = 0.5623E-02	EVI(9) = 0.0000D+00
CI(10) = 0.3720D-02	CM(10) = 0.3617E-02	EVI(10) = 0.0000D+00

TVI(1) = 0.8436E+04
TVI(2) = 0.9134E+04
TVI(3) = 0.0000E+00
TVI(4) = 0.0000E+00
TVI(5) = 0.9146E+04
TVI(6) = 0.9145E+04
TVI(7) = 0.0000E+00
TVI(8) = 0.9193E+04
TVI(9) = 0.0000E+00
TVI(10) = 0.0000E+00 TE1= 0.9037E+04 ITRTE= 3 W1= 0.246E+27

PSI = 6 IX = 451 SHOCK J COND. = 1 CVD MODEL = 4
X = .7723D+01 H = .3930D+12 MU = .1551E+02 P = .1920E+05 RHO = .3969E-06
U = .1450E+05 T = .9022E+04 E = .3446E+12 CIT = .1563E-01 SUMCI = .1000E+01

CI(1) = 0.1508D+00	CM(1) = 0.8348E-01	EVI(1) = 0.2020D+11
CI(2) = 0.3635D-04	CM(2) = 0.1762E-04	EVI(2) = 0.2060D+11
CI(3) = 0.6109D+00	CM(3) = 0.6763E+00	EVI(3) = 0.0000D+00
CI(4) = 0.2269D+00	CM(4) = 0.2200E+00	EVI(4) = 0.0000D+00
CI(5) = 0.2254D-02	CM(5) = 0.1165E-02	EVI(5) = 0.2141D+11
CI(6) = 0.7160D-04	CM(6) = 0.3700E-04	EVI(6) = 0.2057D+11
CI(7) = 0.3367D-06	CM(7) = 0.9518E-02	EVI(7) = 0.0000D+00
CI(8) = 0.1053D-03	CM(8) = 0.5831E-04	EVI(8) = 0.2254D+11
CI(9) = 0.5137D-02	CM(9) = 0.5687E-02	EVI(9) = 0.0000D+00
CI(10) = 0.3854D-02	CM(10) = 0.3736E-02	EVI(10) = 0.0000D+00

TVI(1) = 0.8387E+04
TVI(2) = 0.9017E+04
TVI(3) = 0.0000E+00
TVI(4) = 0.0000E+00
TVI(5) = 0.9028E+04
TVI(6) = 0.9027E+04
TVI(7) = 0.0000E+00
TVI(8) = 0.9070E+04
TVI(9) = 0.0000E+00
TVI(10) = 0.0000E+00 TE1= 0.8936E+04 ITRTE= 3 W1= 0.213E+27

PSI = 6 IX = 501 SHOCK J COND. = 1 CVD MODEL = 4
X = .8504D+01 H = .3929D+12 MU = .1547E+02 P = .1919E+05 RHO = .4000E-06
U = .1570E+05 T = .8926E+04 E = .3450E+12 CIT = .1563E-01 SUMCI = .1000E+01

CI(1) = 0.1466D+00	CM(1) = 0.8097E-01	EVI(1) = 0.2007D+11
CI(2) = 0.3364D-04	CM(2) = 0.1626E-04	EVI(2) = 0.2035D+11
CI(3) = 0.6151D+00	CM(3) = 0.6792E+00	EVI(3) = 0.0000D+00
CI(4) = 0.2269D+00	CM(4) = 0.2194E+00	EVI(4) = 0.0000D+00
CI(5) = 0.2150D-02	CM(5) = 0.1108E-02	EVI(5) = 0.2114D+11
CI(6) = 0.6612D-04	CM(6) = 0.3409E-04	EVI(6) = 0.2031D+11

CI(7) = 0.3425D-06	CM(7) = 0.9659E-02	EVI(7) = 0.0000D+00
CI(8) = 0.9513D-04	CM(8) = 0.5253E-04	EVI(8) = 0.2225D+11
CI(9) = 0.5197D-02	CM(9) = 0.5739E-02	EVI(9) = 0.0000D+00
CI(10) = 0.3964D-02	CM(10) = 0.3833E-02	EVI(10) = 0.0000D+00

TVI(1) = 0.8343E+04	
TVI(2) = 0.8920E+04	
TVI(3) = 0.0000E+00	
TVI(4) = 0.0000E+00	
TVI(5) = 0.8931E+04	
TVI(6) = 0.8930E+04	
TVI(7) = 0.0000E+00	
TVI(8) = 0.8970E+04	
TVI(9) = 0.0000E+00	
TVI(10) = 0.0000E+00	TE1= 0.8852E+04 ITRTE= 3 W1= 0.188E+27

PSI = 6 IX = 551 SHOCK J COND. = 1 CVD MODEL = 4
X = .9754D+01 H = .3929D+12 MU = .1542E+02 P = .1918E+05 RHO = .4040E-06
U = .1762E+05 T = .8804E+04 E = .3454E+12 CIT = .3125E-01 SUMCI = .1000E+01

CI(1) = 0.1413D+00	CM(1) = 0.7778E-01	EVI(1) = 0.1989D+11
CI(2) = 0.3031D-04	CM(2) = 0.1460E-04	EVI(2) = 0.2003D+11
CI(3) = 0.6204D+00	CM(3) = 0.6829E+00	EVI(3) = 0.0000D+00
CI(4) = 0.2268D+00	CM(4) = 0.2186E+00	EVI(4) = 0.0000D+00
CI(5) = 0.2014D-02	CM(5) = 0.1035E-02	EVI(5) = 0.2081D+11
CI(6) = 0.5966D-04	CM(6) = 0.3066E-04	EVI(6) = 0.1997D+11
CI(7) = 0.3500D-06	CM(7) = 0.9838E-02	EVI(7) = 0.0000D+00
CI(8) = 0.8334D-04	CM(8) = 0.4587E-04	EVI(8) = 0.2187D+11
CI(9) = 0.5274D-02	CM(9) = 0.5805E-02	EVI(9) = 0.0000D+00
CI(10) = 0.4105D-02	CM(10) = 0.3956E-02	EVI(10) = 0.0000D+00

TVI(1) = 0.8282E+04	
TVI(2) = 0.8798E+04	
TVI(3) = 0.0000E+00	
TVI(4) = 0.0000E+00	
TVI(5) = 0.8808E+04	
TVI(6) = 0.8807E+04	
TVI(7) = 0.0000E+00	
TVI(8) = 0.8843E+04	
TVI(9) = 0.0000E+00	
TVI(10) = 0.0000E+00	TE1= 0.8743E+04 ITRTE= 3 W1= 0.159E+27

PSI = 6 IX = 559 SHOCK J COND. = 1 CVD MODEL = 4
X = .1000D+02 H = .3929D+12 MU = .1541E+02 P = .1917E+05 RHO = .4047E-06
U = .1800E+05 T = .8783E+04 E = .3455E+12 CIT = .3125E-01 SUMCI = .1000E+01

CI(1) = 0.1404D+00	CM(1) = 0.7724E-01	EVI(1) = 0.1986D+11
CI(2) = 0.2975D-04	CM(2) = 0.1433E-04	EVI(2) = 0.1998D+11
CI(3) = 0.6213D+00	CM(3) = 0.6835E+00	EVI(3) = 0.0000D+00
CI(4) = 0.2268D+00	CM(4) = 0.2185E+00	EVI(4) = 0.0000D+00
CI(5) = 0.1991D-02	CM(5) = 0.1023E-02	EVI(5) = 0.2075D+11
CI(6) = 0.5861D-04	CM(6) = 0.3010E-04	EVI(6) = 0.1991D+11
CI(7) = 0.3513D-06	CM(7) = 0.9868E-02	EVI(7) = 0.0000D+00
CI(8) = 0.8144D-04	CM(8) = 0.4480E-04	EVI(8) = 0.2181D+11
CI(9) = 0.5287D-02	CM(9) = 0.5816E-02	EVI(9) = 0.0000D+00
CI(10) = 0.4129D-02	CM(10) = 0.3977E-02	EVI(10) = 0.0000D+00

TVI(1) = 0.8272E+04	
TVI(2) = 0.8777E+04	
TVI(3) = 0.0000E+00	
TVI(4) = 0.0000E+00	
TVI(5) = 0.8787E+04	
TVI(6) = 0.8786E+04	
TVI(7) = 0.0000E+00	
TVI(8) = 0.8821E+04	
TVI(9) = 0.0000E+00	
TVI(10) = 0.0000E+00	TE1= 0.8725E+04 ITRTE= 3 W1= 0.154E+27

XXXXXXXXXXXXXXXXXX

X = .1000D+02 H = .3924D+12 MU = .1900E+02 P = .1919E+05 RHO = .2605E-06
 U = .3580E+05 T = .1683E+05 E = .3188E+12 CIT = .3125E-01 SUMCI = .1000E+01

CI(1) = 0.4615D+00	CM(1) = 0.3130E+00	EVI(1) = 0.2556D+11
CI(2) = 0.8119D-03	CM(2) = 0.4821E-03	EVI(2) = 0.3978D+11
CI(3) = 0.2967D+00	CM(3) = 0.4025E+00	EVI(3) = 0.0000D+00
CI(4) = 0.2259D+00	CM(4) = 0.2682E+00	EVI(4) = 0.0000D+00
CI(5) = 0.8527D-02	CM(5) = 0.5399E-02	EVI(5) = 0.4339D+11
CI(6) = 0.1492D-02	CM(6) = 0.9448E-03	EVI(6) = 0.4242D+11
CI(7) = 0.1502D-06	CM(7) = 0.5201E-02	EVI(7) = 0.0000D+00
CI(8) = 0.4173D-02	CM(8) = 0.2830E-02	EVI(8) = 0.4708D+11
CI(9) = 0.1011D-02	CM(9) = 0.1371E-02	EVI(9) = 0.0000D+00
CI(10) = 0.4616D-04	CM(10) = 0.5481E-04	EVI(10) = 0.0000D+00

TVI(1) = 0.1022E+05
 TVI(2) = 0.1642E+05
 TVI(3) = 0.0000E+00
 TVI(4) = 0.0000E+00
 TVI(5) = 0.1699E+05
 TVI(6) = 0.1696E+05
 TVI(7) = 0.0000E+00
 TVI(8) = 0.1738E+05
 TVI(9) = 0.0000E+00
 TVI(10) = 0.0000E+00 TE1= 0.1138E+05 ITRTE= 5 W1= 0.615E+28

XXXXXXXXXXXXXXXXXX

PSI = 22 IX = 1 SHOCK J COND. = 1 CVD MODEL = 4
 X = .9500D+01 H = .3924D+12 MU = .2885E+02 P = .1920E+05 RHO = .2195E-06
 U = .3666E+05 T = .3035E+05 E = .3049E+12 CIT = .1221E-03 SUMCI = .1000E+01

CI(1) = 0.7680D+00	CM(1) = 0.7908E+00	EVI(1) = 0.3378D+03
CI(2) = 0.2320D+00	CM(2) = 0.2092E+00	EVI(2) = 0.5740D+05
CI(3) = 0.0000D+00	CM(3) = 0.0000E+00	EVI(3) = 0.0000D+00
CI(4) = 0.0000D+00	CM(4) = 0.0000E+00	EVI(4) = 0.0000D+00
CI(5) = 0.0000D+00	CM(5) = 0.0000E+00	EVI(5) = 0.6999D+04
CI(6) = 0.0000D+00	CM(6) = 0.0000E+00	EVI(6) = 0.2747D+03
CI(7) = 0.0000D+00	CM(7) = 0.0000E+00	EVI(7) = 0.0000D+00
CI(8) = 0.0000D+00	CM(8) = 0.0000E+00	EVI(8) = 0.1184D+04
CI(9) = 0.0000D+00	CM(9) = 0.0000E+00	EVI(9) = 0.0000D+00
CI(10) = 0.0000D+00	CM(10) = 0.0000E+00	EVI(10) = 0.0000D+00

TVI(1) = 0.1971E+03
 TVI(2) = 0.1971E+03
 TVI(3) = 0.0000E+00
 TVI(4) = 0.0000E+00
 TVI(5) = 0.1971E+03
 TVI(6) = 0.1971E+03
 TVI(7) = 0.0000E+00
 TVI(8) = 0.1971E+03
 TVI(9) = 0.0000E+00
 TVI(10) = 0.0000E+00 TE1= 0.1971E+03 ITRTE= 5 W1= 0.615E+28

BEGIN PHYSICAL SPACE CALCULATIONS KSTAG = 209

PSI = 0.00000E+00
 X = 0.500000E+00 R = 0.466921E+00 Y = 0.152165E+02 Z = 0.152170E+02

PSI = 0.30828E-02
 X = 0.500000E+00 R = 0.500000E+00 Y = 0.000000E+00 Z = 0.559944E-03

PSI = 0.00000E+00
 X = 0.100000E+01 R = 0.935523E+00 Y = 0.148298E+02 Z = 0.148318E+02

PSI = 0.30828E-02
 X = 0.100000E+01 R = 0.953088E+00 Y = 0.107896E+02 Z = 0.107917E+02

PSI = 0.12331E-01
 X = 0.100000E+01 R = 0.100052E+01 Y = -0.119472E+00 Z = -0.117292E+00

X = 0.750000E+01	R = 0.712156E+01	Y = 0.115977E+02	Z = 0.117137E+02
PSI = 0.15105E+00			
X = 0.750000E+01	R = 0.715026E+01	Y = 0.107152E+02	Z = 0.108316E+02
PSI = 0.19728E+00			
X = 0.750000E+01	R = 0.718215E+01	Y = 0.973441E+01	Z = 0.985138E+01
PSI = 0.24968E+00			
X = 0.750000E+01	R = 0.721706E+01	Y = 0.866062E+01	Z = 0.877815E+01
PSI = 0.30823E+00			
X = 0.750000E+01	R = 0.725492E+01	Y = 0.749644E+01	Z = 0.761459E+01
PSI = 0.37295E+00			
X = 0.750000E+01	R = 0.729573E+01	Y = 0.624120E+01	Z = 0.636001E+01
PSI = 0.44382E+00			
X = 0.750000E+01	R = 0.733970E+01	Y = 0.488914E+01	Z = 0.500867E+01
PSI = 0.52086E+00			
X = 0.750000E+01	R = 0.738729E+01	Y = 0.342541E+01	Z = 0.354571E+01
PSI = 0.60404E+00			
X = 0.750000E+01	R = 0.743965E+01	Y = 0.181530E+01	Z = 0.193645E+01
PSI = 0.69339E+00			
X = 0.750000E+01	R = 0.749868E+01	Y = 0.721238E-04	Z = 0.122182E+00
PSI = 0.00000E+00			
X = 0.800000E+01	R = 0.750273E+01	Y = 0.142962E+02	Z = 0.144265E+02
PSI = 0.30828E-02			
X = 0.800000E+01	R = 0.750523E+01	Y = 0.142242E+02	Z = 0.143546E+02
PSI = 0.12331E-01			
X = 0.800000E+01	R = 0.751266E+01	Y = 0.140098E+02	Z = 0.141403E+02
PSI = 0.27745E-01			
X = 0.800000E+01	R = 0.752478E+01	Y = 0.136602E+02	Z = 0.137909E+02
PSI = 0.49324E-01			
X = 0.800000E+01	R = 0.754124E+01	Y = 0.131857E+02	Z = 0.133167E+02
PSI = 0.77067E-01			
X = 0.800000E+01	R = 0.756165E+01	Y = 0.125970E+02	Z = 0.127283E+02
PSI = 0.11097E+00			
X = 0.800000E+01	R = 0.758567E+01	Y = 0.119042E+02	Z = 0.120359E+02
PSI = 0.15105E+00			
X = 0.800000E+01	R = 0.761300E+01	Y = 0.111157E+02	Z = 0.112479E+02
PSI = 0.19728E+00			
X = 0.800000E+01	R = 0.764342E+01	Y = 0.102383E+02	Z = 0.103711E+02
PSI = 0.24968E+00			
X = 0.800000E+01	R = 0.767675E+01	Y = 0.927721E+01	Z = 0.941054E+01
PSI = 0.30823E+00			
X = 0.800000E+01	R = 0.771286E+01	Y = 0.823566E+01	Z = 0.836962E+01
PSI = 0.37295E+00			
X = 0.800000E+01	R = 0.775172E+01	Y = 0.711489E+01	Z = 0.724952E+01
PSI = 0.44382E+00			
X = 0.800000E+01	R = 0.779338E+01	Y = 0.591316E+01	Z = 0.604851E+01
PSI = 0.52086E+00			
X = 0.800000E+01	R = 0.783807E+01	Y = 0.462417E+01	Z = 0.476030E+01
PSI = 0.60404E+00			
X = 0.800000E+01	R = 0.788629E+01	Y = 0.323339E+01	Z = 0.337035E+01
PSI = 0.69339E+00			
X = 0.800000E+01	R = 0.793915E+01	Y = 0.170879E+01	Z = 0.184667E+01

PSI = 0.78888E+00
X = 0.800000E+01 R = 0.799839E+01 Y = 0.377804E-04 Z = 0.138945E+00

PSI = 0.00000E+00
X = 0.850000E+01 R = 0.797149E+01 Y = 0.143007E+02 Z = 0.144478E+02

PSI = 0.30828E-02
X = 0.850000E+01 R = 0.797383E+01 Y = 0.142369E+02 Z = 0.143841E+02

PSI = 0.12331E-01
X = 0.850000E+01 R = 0.798084E+01 Y = 0.140466E+02 Z = 0.141939E+02

PSI = 0.27745E-01
X = 0.850000E+01 R = 0.799230E+01 Y = 0.137355E+02 Z = 0.138830E+02

PSI = 0.49324E-01
X = 0.850000E+01 R = 0.800789E+01 Y = 0.133120E+02 Z = 0.134597E+02

PSI = 0.77067E-01
X = 0.850000E+01 R = 0.802730E+01 Y = 0.127851E+02 Z = 0.129332E+02

PSI = 0.11097E+00
X = 0.850000E+01 R = 0.805019E+01 Y = 0.121634E+02 Z = 0.123120E+02

PSI = 0.15105E+00
X = 0.850000E+01 R = 0.807629E+01 Y = 0.114546E+02 Z = 0.116036E+02

PSI = 0.19728E+00
X = 0.850000E+01 R = 0.810537E+01 Y = 0.106647E+02 Z = 0.108143E+02

PSI = 0.24968E+00
X = 0.850000E+01 R = 0.813726E+01 Y = 0.979873E+01 Z = 0.994887E+01

PSI = 0.30823E+00
X = 0.850000E+01 R = 0.817182E+01 Y = 0.886015E+01 Z = 0.901092E+01

PSI = 0.37295E+00
X = 0.850000E+01 R = 0.820898E+01 Y = 0.785104E+01 Z = 0.800250E+01

PSI = 0.44382E+00
X = 0.850000E+01 R = 0.824873E+01 Y = 0.677160E+01 Z = 0.692379E+01

PSI = 0.52086E+00
X = 0.850000E+01 R = 0.829116E+01 Y = 0.561938E+01 Z = 0.577235E+01

PSI = 0.60404E+00
X = 0.850000E+01 R = 0.833651E+01 Y = 0.438777E+01 Z = 0.454158E+01

PSI = 0.69339E+00
X = 0.850000E+01 R = 0.838530E+01 Y = 0.306281E+01 Z = 0.321752E+01

PSI = 0.78888E+00
X = 0.850000E+01 R = 0.843862E+01 Y = 0.161463E+01 Z = 0.177032E+01

PSI = 0.89053E+00
X = 0.850000E+01 R = 0.849808E+01 Y = 0.593820E-04 Z = 0.156840E+00

PSI = 0.00000E+00
X = 0.900000E+01 R = 0.844020E+01 Y = 0.143059E+02 Z = 0.144707E+02

PSI = 0.30828E-02
X = 0.900000E+01 R = 0.844242E+01 Y = 0.142489E+02 Z = 0.144138E+02

PSI = 0.12331E-01
X = 0.900000E+01 R = 0.844904E+01 Y = 0.140788E+02 Z = 0.142439E+02

PSI = 0.27745E-01
X = 0.900000E+01 R = 0.845990E+01 Y = 0.138002E+02 Z = 0.139654E+02

PSI = 0.49324E-01
X = 0.900000E+01 R = 0.847473E+01 Y = 0.134198E+02 Z = 0.135854E+02

PSI = 0.77067E-01
X = 0.900000E+01 R = 0.849321E+01 Y = 0.129455E+02 Z = 0.131114E+02

PSI = 0.11097E+00
 X = 0.900000E+01 R = 0.851507E+01 Y = 0.123847E+02 Z = 0.125510E+02

PSI = 0.15105E+00
 X = 0.900000E+01 R = 0.854004E+01 Y = 0.117439E+02 Z = 0.119107E+02

PSI = 0.19728E+00
 X = 0.900000E+01 R = 0.856791E+01 Y = 0.110288E+02 Z = 0.111961E+02

PSI = 0.24968E+00
 X = 0.900000E+01 R = 0.859849E+01 Y = 0.102440E+02 Z = 0.104120E+02

PSI = 0.30823E+00
 X = 0.900000E+01 R = 0.863165E+01 Y = 0.939311E+01 Z = 0.956170E+01

PSI = 0.37295E+00
 X = 0.900000E+01 R = 0.866730E+01 Y = 0.847844E+01 Z = 0.864773E+01

PSI = 0.44382E+00
 X = 0.900000E+01 R = 0.870539E+01 Y = 0.750118E+01 Z = 0.767121E+01

PSI = 0.52086E+00
 X = 0.900000E+01 R = 0.874593E+01 Y = 0.646079E+01 Z = 0.663161E+01

PSI = 0.60404E+00
 X = 0.900000E+01 R = 0.878905E+01 Y = 0.535434E+01 Z = 0.552601E+01

PSI = 0.69339E+00
 X = 0.900000E+01 R = 0.883501E+01 Y = 0.417518E+01 Z = 0.434774E+01

PSI = 0.78888E+00
 X = 0.900000E+01 R = 0.888432E+01 Y = 0.290976E+01 Z = 0.308328E+01

PSI = 0.89053E+00
 X = 0.900000E+01 R = 0.893808E+01 Y = 0.153036E+01 Z = 0.170492E+01

PSI = 0.99832E+00
 X = 0.900000E+01 R = 0.899772E+01 Y = 0.377593E-04 Z = 0.175767E+00

PSI = 0.00000E+00
 X = 0.950000E+01 R = 0.890883E+01 Y = 0.143122E+02 Z = 0.144959E+02

PSI = 0.30828E-02
 X = 0.950000E+01 R = 0.891094E+01 Y = 0.142611E+02 Z = 0.144448E+02

PSI = 0.12331E-01
 X = 0.950000E+01 R = 0.891722E+01 Y = 0.141081E+02 Z = 0.142919E+02

PSI = 0.27745E-01
 X = 0.950000E+01 R = 0.892755E+01 Y = 0.138571E+02 Z = 0.140411E+02

PSI = 0.49324E-01
 X = 0.950000E+01 R = 0.894167E+01 Y = 0.135137E+02 Z = 0.136980E+02

PSI = 0.77067E-01
 X = 0.950000E+01 R = 0.895932E+01 Y = 0.130844E+02 Z = 0.132691E+02

PSI = 0.11097E+00
 X = 0.950000E+01 R = 0.898023E+01 Y = 0.125758E+02 Z = 0.127609E+02

PSI = 0.15105E+00
 X = 0.950000E+01 R = 0.900417E+01 Y = 0.119936E+02 Z = 0.121792E+02

PSI = 0.19728E+00
 X = 0.950000E+01 R = 0.903092E+01 Y = 0.113430E+02 Z = 0.115291E+02

PSI = 0.24968E+00
 X = 0.950000E+01 R = 0.906031E+01 Y = 0.106282E+02 Z = 0.108149E+02

PSI = 0.30823E+00
 X = 0.950000E+01 R = 0.909220E+01 Y = 0.985261E+01 Z = 0.100400E+02

PSI = 0.37295E+00
 X = 0.950000E+01 R = 0.912648E+01 Y = 0.901885E+01 Z = 0.920697E+01

PSI = 0.44382E+00

X = 0.950000E+01 R = 0.916310E+01 Y = 0.812848E+01 Z = 0.831735E+01
 PSI = 0.52086E+00
 X = 0.950000E+01 R = 0.920201E+01 Y = 0.718196E+01 Z = 0.737164E+01
 PSI = 0.60404E+00
 X = 0.950000E+01 R = 0.924328E+01 Y = 0.617825E+01 Z = 0.636878E+01
 PSI = 0.69339E+00
 X = 0.950000E+01 R = 0.928704E+01 Y = 0.511414E+01 Z = 0.530557E+01
 PSI = 0.78888E+00
 X = 0.950000E+01 R = 0.933355E+01 Y = 0.398290E+01 Z = 0.417528E+01
 PSI = 0.89053E+00
 X = 0.950000E+01 R = 0.938336E+01 Y = 0.277153E+01 Z = 0.296494E+01
 PSI = 0.99832E+00
 X = 0.950000E+01 R = 0.943751E+01 Y = 0.145455E+01 Z = 0.164907E+01
 PSI = 0.11123E+01
 X = 0.950000E+01 R = 0.949732E+01 Y = 0.489373E-04 Z = 0.195801E+00

OLSTAD RADIATION MODEL

BETA=BETA(Te)

AT X = 0.1500D+01

FOR BAND 1 QR = 0.8778D-01 WATTS/SQ.CM.
 FOR BAND 2 QR = 0.7180D-01 WATTS/SQ.CM.
 FOR BAND 3 QR = 0.1933D+00 WATTS/SQ.CM.
 FOR BAND 4 QR = 0.9876D-01 WATTS/SQ.CM.
 FOR BAND 5 QR = 0.4015D-01 WATTS/SQ.CM.
 FOR BAND 6 QR = 0.3330D+00 WATTS/SQ.CM.
 FOR BAND 7 QR = 0.5897D-09 WATTS/SQ.CM.
 FOR BAND 8 QR = 0.5748D+00 WATTS/SQ.CM.
 TOTAL QR = 0.1400D+01 WATTS/SQ.CM.

AT X = 0.2000D+01

FOR BAND 1 QR = 0.9265D-01 WATTS/SQ.CM.
 FOR BAND 2 QR = 0.8536D-01 WATTS/SQ.CM.
 FOR BAND 3 QR = 0.2305D+00 WATTS/SQ.CM.
 FOR BAND 4 QR = 0.1194D+00 WATTS/SQ.CM.
 FOR BAND 5 QR = 0.4887D-01 WATTS/SQ.CM.
 FOR BAND 6 QR = 0.4038D+00 WATTS/SQ.CM.
 FOR BAND 7 QR = 0.3688D-05 WATTS/SQ.CM.
 FOR BAND 8 QR = 0.6846D+00 WATTS/SQ.CM.
 TOTAL QR = 0.1665D+01 WATTS/SQ.CM.

AT X = 0.2500D+01

AT X = 0.8000D+01

FOR BAND 1 QR = 0.3888D+00 WATTS/SQ.CM.

FOR BAND 2 QR = 0.6891D+00 WATTS/SQ.CM.

FOR BAND 3 QR = 0.1734D+01 WATTS/SQ.CM.

FOR BAND 4 QR = 0.8759D+00 WATTS/SQ.CM.

FOR BAND 5 QR = 0.4096D+00 WATTS/SQ.CM.

FOR BAND 6 QR = 0.2517D+01 WATTS/SQ.CM.

FOR BAND 7 QR = 0.2719D-02 WATTS/SQ.CM.

FOR BAND 8 QR = 0.2690D+01 WATTS/SQ.CM.

TOTAL QR = 0.9307D+01 WATTS/SQ.CM.

AT X = 0.8500D+01

FOR BAND 1 QR = 0.4131D+00 WATTS/SQ.CM.

FOR BAND 2 QR = 0.7440D+00 WATTS/SQ.CM.

FOR BAND 3 QR = 0.1846D+01 WATTS/SQ.CM.

FOR BAND 4 QR = 0.9296D+00 WATTS/SQ.CM.

FOR BAND 5 QR = 0.4412D+00 WATTS/SQ.CM.

FOR BAND 6 QR = 0.2681D+01 WATTS/SQ.CM.

FOR BAND 7 QR = 0.2757D-02 WATTS/SQ.CM.

FOR BAND 8 QR = 0.2814D+01 WATTS/SQ.CM.

TOTAL QR = 0.9872D+01 WATTS/SQ.CM.

AT X = 0.9000D+01

FOR BAND 1 QR = 0.4364D+00 WATTS/SQ.CM.

FOR BAND 2 QR = 0.7967D+00 WATTS/SQ.CM.

FOR BAND 3 QR = 0.1952D+01 WATTS/SQ.CM.

FOR BAND 4 QR = 0.9800D+00 WATTS/SQ.CM.

FOR BAND 5 QR = 0.4715D+00 WATTS/SQ.CM.

FOR BAND 6 QR = 0.2835D+01 WATTS/SQ.CM.

FOR BAND 7 QR = 0.2786D-02 WATTS/SQ.CM.

FOR BAND 8 QR = 0.2928D+01 WATTS/SQ.CM.

TOTAL QR = 0.1040D+02 WATTS/SQ.CM.

AT X = 0.9500D+01

FOR BAND 1 QR = 0.4593D+00 WATTS/SQ.CM.

FOR BAND 2 QR = 0.8483D+00 WATTS/SQ.CM.

FOR BAND 3 QR = 0.2057D+01 WATTS/SQ.CM.

FOR BAND 4 QR = 0.1029D+01 WATTS/SQ.CM.

FOR BAND 5 QR = 0.5013D+00 WATTS/SQ.CM.

FOR BAND 6 QR = 0.2986D+01 WATTS/SQ.CM.